UNIVERSITY OF
ILLINOIS LIBRARY
AT URBANA-CHAMPAIGN
STACKS

STRUCTURED SYSTEMS ANALYSIS & DESIGN

Doc.
D101.2:
St8/2

BOOKSTACKS
DOCUMENTS

US ARMY
MANAGEMENT ENGINEERING
TRAINING ACTIVITY
ROCK ISLAND
ILLINOIS

DEPOSITORY
OCT 19 1982
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

FOREWORD

The user of these materials should be aware that they have been developed in conjunction with other items by our staff for use in resident courses at the U. S. Army Management Engineering Training Activity.

Inquiries regarding these materials should be addressed to: Director, U. S. Army Management Engineering Training Activity, Rock Island, Illinois 61299.

Doc.
D101.2:
St 8/2

October 1982

CONTENTS

	SLIDE NUMBER
SYSTEMS ANALYSIS - INTRODUCTION	1 - 16
STRUCTURED SYSTEMS ANALYSIS	17 - 34
DATA FLOW GRAPHS	35 - 60
DATA DICTIONARY	61 - 76
STRUCTURED ENGLISH	77 - 89
DECISION TABLES / TREES	90 - 92
FUNCTIONAL DESCRIPTION	93 - 95
SYSTEM DESIGN - INTRODUCTION	96 - 106
STRATEGIES	107 - 113
MODULAR STRUCTURES	114 - 122
STRUCTURE CHARTS	123 - 130
COUPLING & COHESION	131 - 148
MORPHOLOGY & HEURISTICS	149 - 161
DESIGN METHODS	162 - 199

OBJECTIVES - SCOPE - COVERAGE

- OBJECTIVE

  ESTABLISH A STRUCTURED APPROACH TO THE DEVELOPMENT OF CORRECT SYSTEMS

- SCOPE

  ANALYSIS - PRESENT SYSTEM

  DESIGN - NEW AUTOMATED SYSTEM

- COVERAGE

  MANAGEMENT

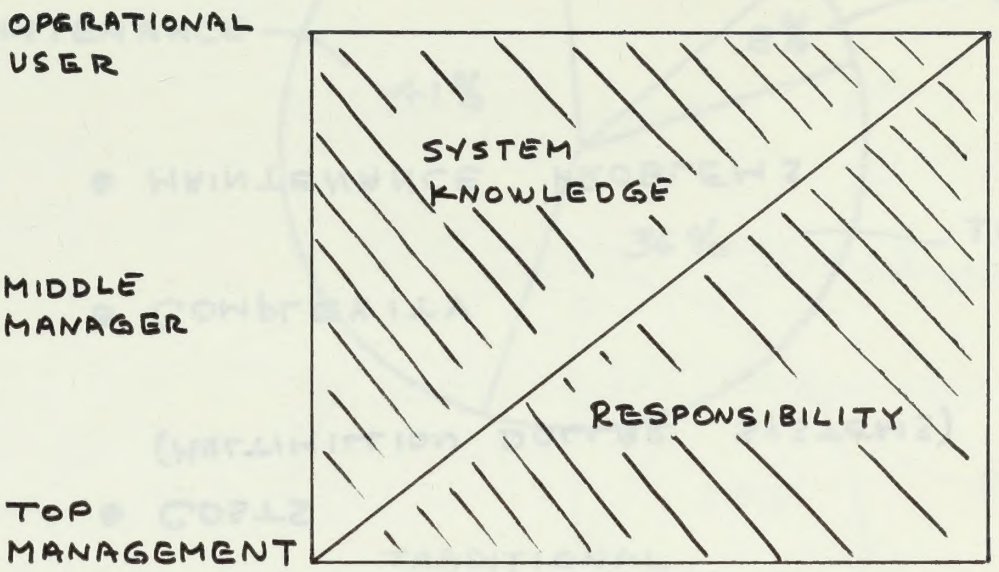
  FUNCTIONAL SPECIALISTS

  DATA PROCESSORS

CHARACTERISTICS OF STRUCTURED APPROACH

- LIFE CYCLE ORIENTED
- STRUCTURED
- CONTROLLED
- USER ORIENTED
  - TOP MANAGEMENT
  - MIDDLE MANAGEMENT
  - OPERATIONAL USER

SYSTEM KNOWLEDGE

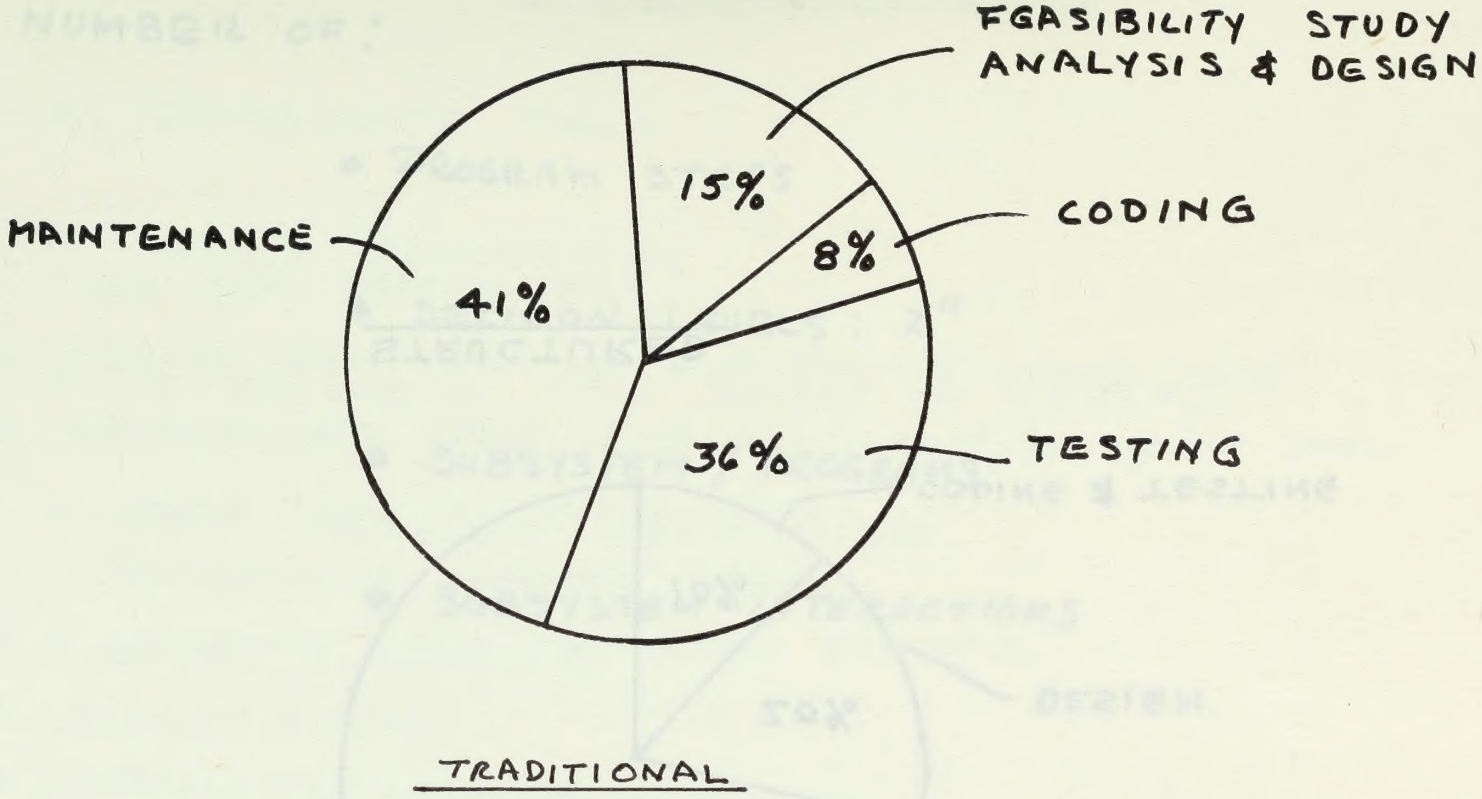


NEED FOR NEW APPROACH

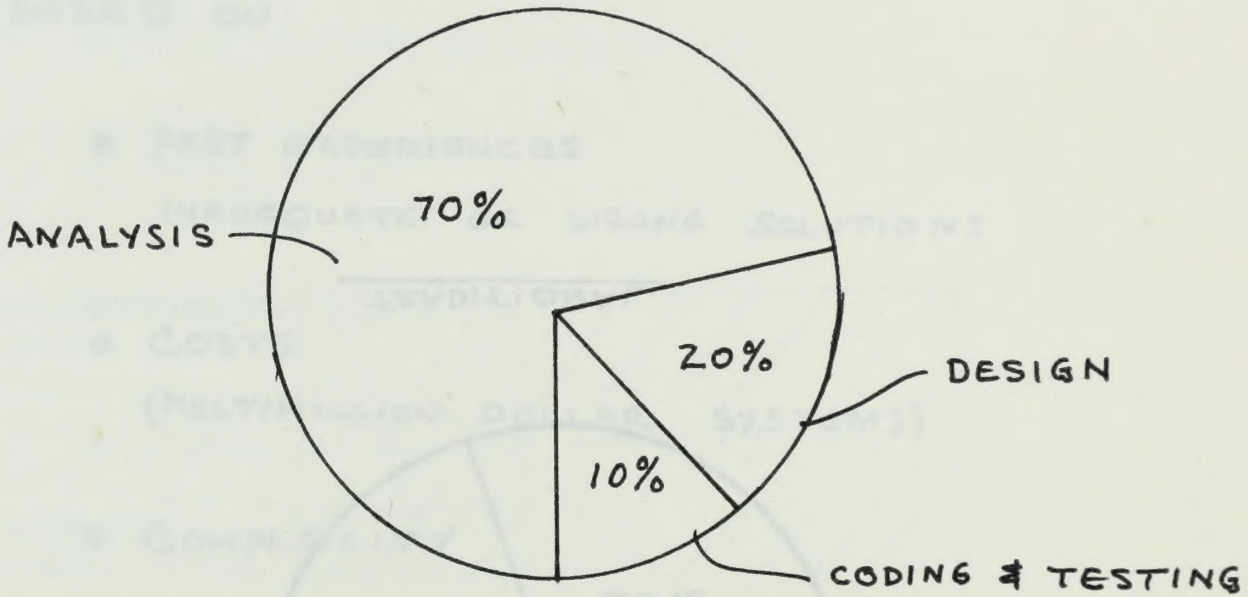
BASED ON:

- PAST EXPERIENCES
  INADEQUATE OR WRONG SOLUTIONS
- COSTS
  (MULTIMILLION DOLLAR SYSTEMS)
- COMPLEXITY
- MAINTENANCE PROBLEMS

EXPENDITURE OF RESOURCES



TRADITIONAL



STRUCTURED

COMPLEXITY MEASURES

BASED ON
NUMBER OF:

- PROGRAM STEPS
- DECISION LEVELS : 2^N
- SUBSYSTEM / PROGRAMS
- SUBSYSTEM INTERACTIONS

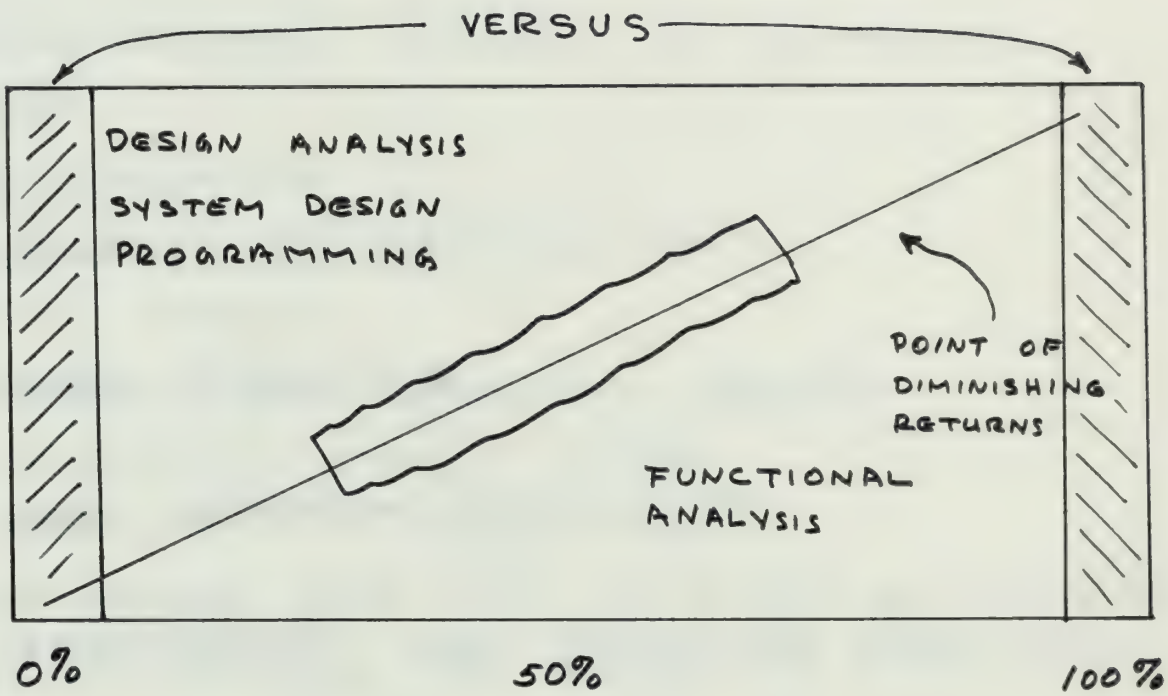
CURRENT MAINTENANCE PROBLEMS

- SYSTEM/PROGRAMS DIFFICULT TO COMPREHEND
  - SIZE
  - DOCUMENTATION
- COSTLY MAINTENANCE
- SYSTEM ERRORS
- PROGRAMS NEVER COMPLETELY DEBUGGED
  - MAINTENANCE INDUCED ERRORS
  - ERRORS OCCUR AT WORST TIME
- MAINTENANCE COSTS 50% TO > 100% OF INITIAL COST
- USE OF RESOURCES FOR MAINTENANCE LEAVES LESS FOR NEW DEVELOPMENT

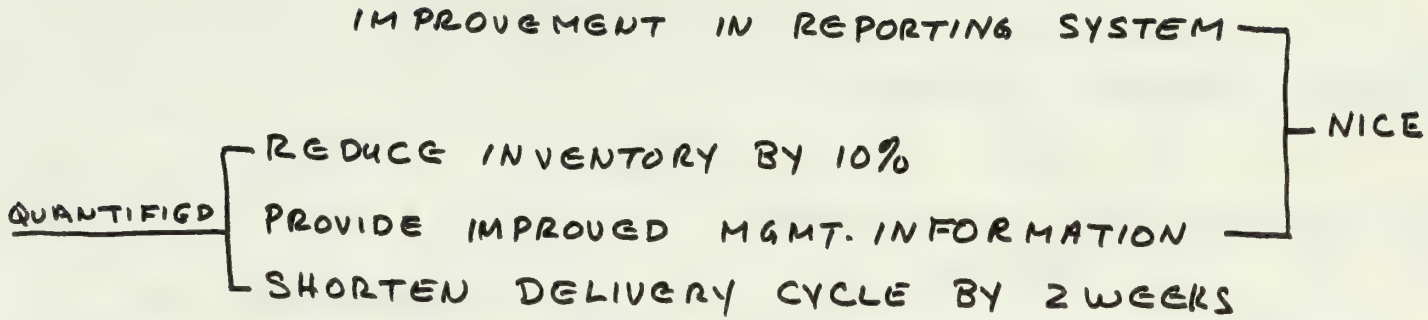
BASIC PROBLEMS

- INADEQUATE TOOLS
- WORLD DOMINATION
- INADEQUATE TIME SPENT ON ANALYSIS
- POOR PROBLEM DEFINITION
- POOR PLANNING
- LACK OF COMMUNICATION

ALLOCATION OF RESOURCES



OBJECTIVES



LEVELS OF OBJECTIVES

OVERALL - DEVISE AN MIS

DETAIL - PROVIDE X-STATUS DAILY

PLANNING

DO IT :

- EARLY

- CONTINUOUSLY

- BRAVELY

- WITH COMMON SENSE

- VISIBLY

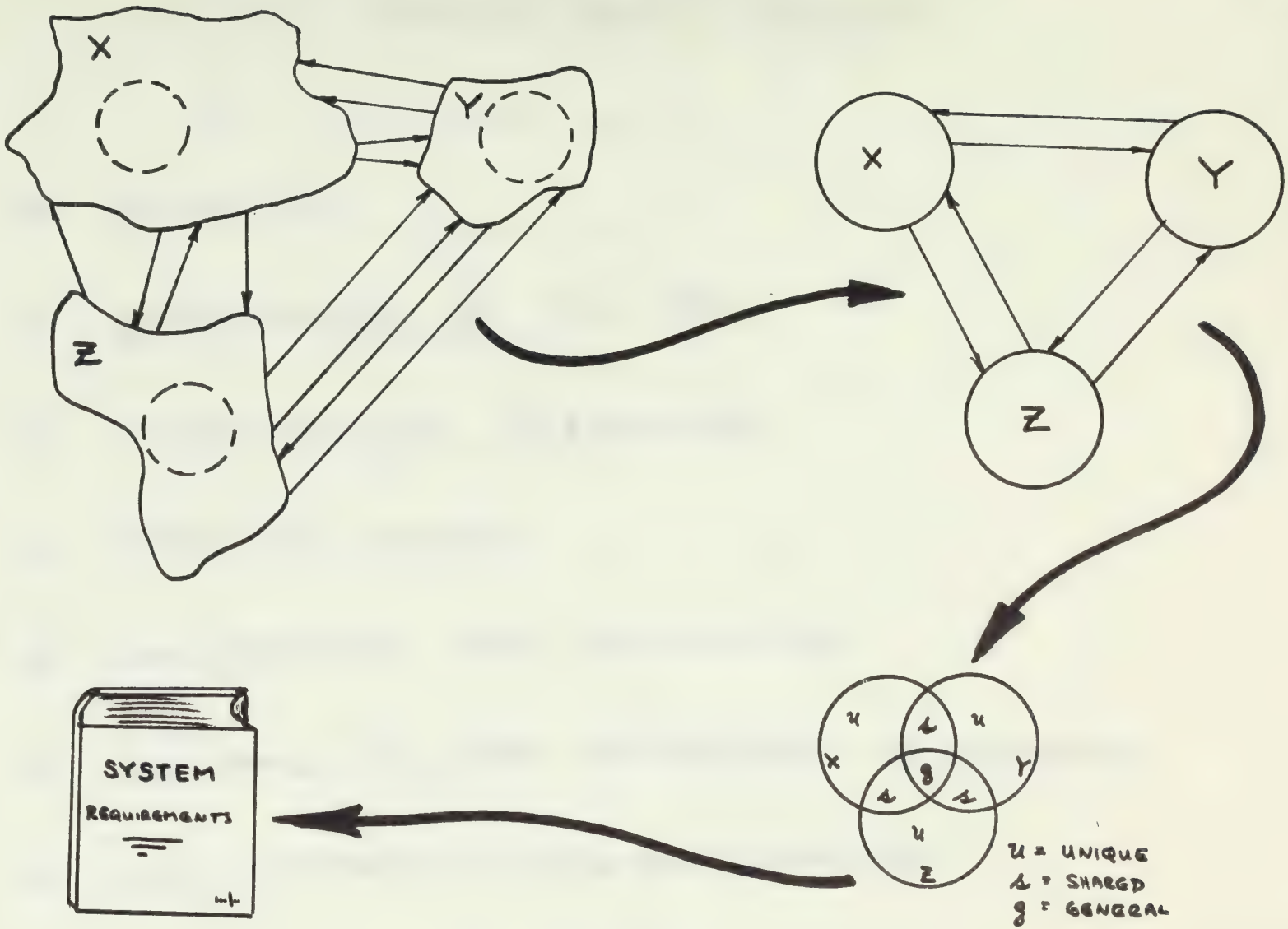
- SIMPLY

KEEP IT SIMPLE STUPID (KISS)

OR

ASSURE SYSTEM SIMPLICITY

ANALYSIS


X
Y
Z
X
Y
Z
u
s
u
X
Y
g
s
s
u
Z
u = UNIQUE
s = SHARED
g = GENERAL
SYSTEM
REQUIREMENTS

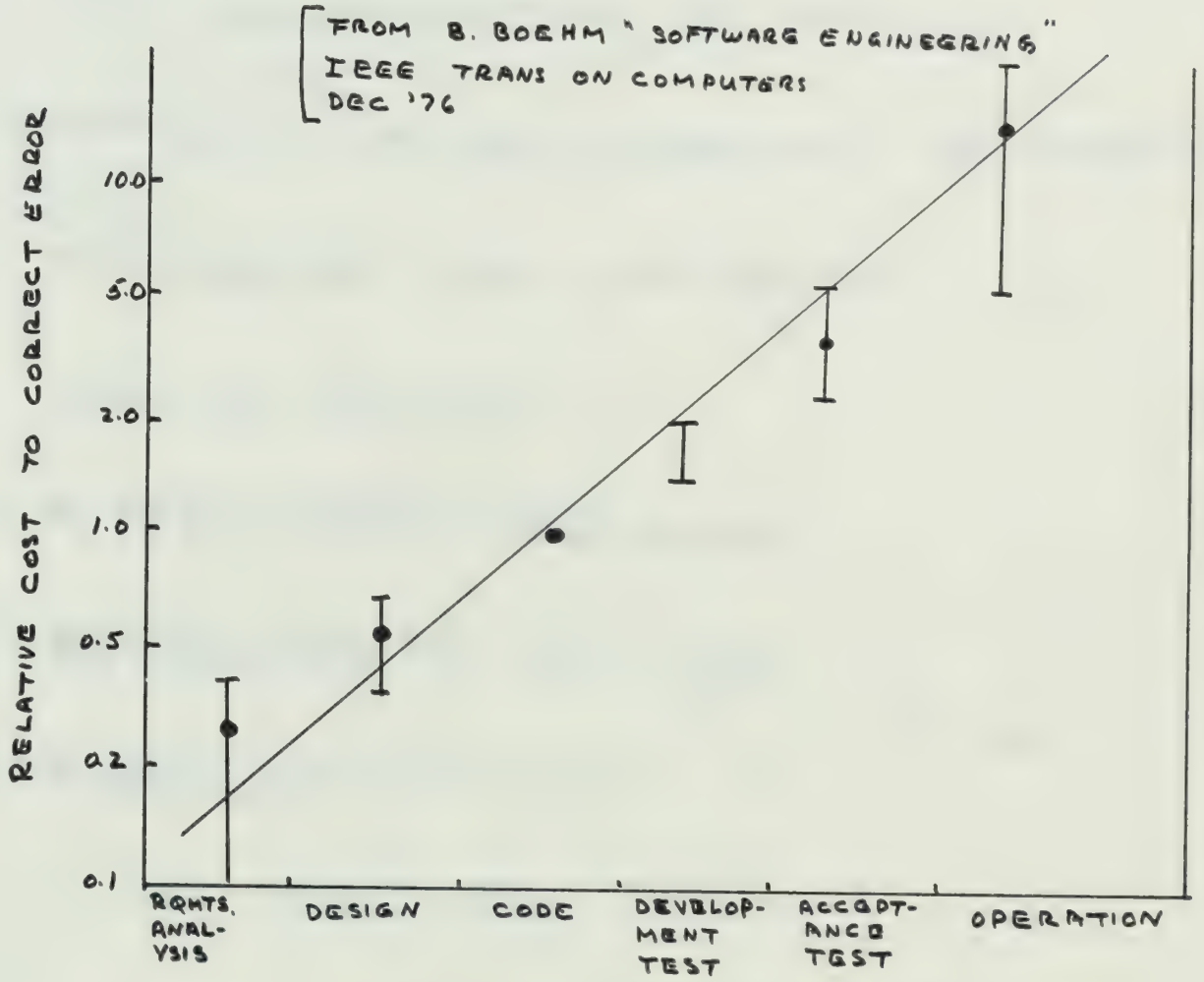
LACK OF COMMUNICATION

- POOR DOCUMENTATION OF PROBLEM
- DIFFICULTY OF IDEA EXPRESSION W/O AMBIGUITY
- MANAGEMENT NON-INVOLVEMENT
- SIZE OF EFFORT
- INTERACTIONS UNDEFINED
- RESISTANCE TO CHANGE
- OMISSIONS

CURRENT SYSTEM PROBLEMS

- HIGH DEVELOPMENT COSTS
- HIGH MAINTENANCE
- LOW RELIABILITY
- LOW CREDIBILITY

ERROR CORRECTION COSTS



BENEFITS OF A STRUCTURED APPROACH

- MAKES USE OF SCIENTIFIC METHOD
- CAN BE USED ON ANY KIND/LEVEL SYSTEM
- MODULAR FOR PROBLEM SOLUTION/MAINTENANCE
- DOCUMENTS THE PROCESS
- PROMOTES VISIBILITY
- IMPROVES COMMUNICATION

CHARACTERISTICS

"PROGRAMMING" VS ANALYSIS

"PROGRAMMING"	ANALYSIS
• STRAIGHTFORWARD - FINAL PRODUCT	• AMORPHOUS
• FEW INTERPERSONAL RELATIONS - MAN/MACHINE	• MANY, COMPLEX RELATIONSHIPS
• DEFINITIVE BODY OF KNOWLEDGE MANUALS, TECHNIQUES	• NON-DEFINITIVE
• SATISFYING - DEFINITE SOLUTION, HARD FACTS	• OFTEN LESS THAN SATISFYING - SOLUTION MAY BE ELUSIVE

SCIENTIFIC METHOD *

(SEVEN PILLARS OF WISDOM)

1. DETERMINE OBJECTIVES

2. ANALYZE THE PROBLEM

3. EVALUATION

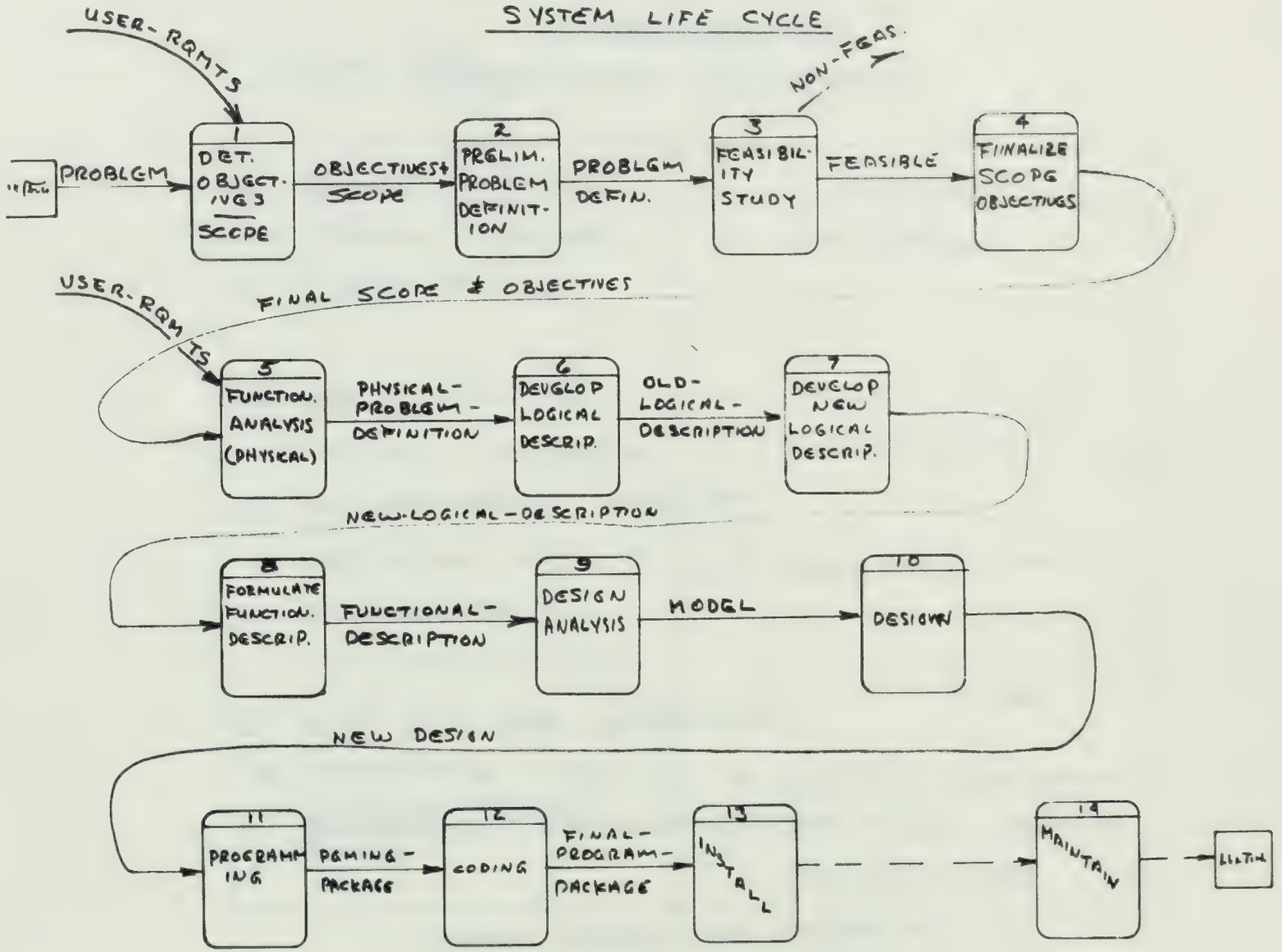
4. IMPROVEMENT - SYNTHESIS

5. TEST

6. TRIAL

7. APPLY

* CIRCA 1950


SYSTEM LIFE CYCLE
USER-RQMTS
PROBLEM
1
DET. OBJECTIVES
SCOPE
OBJECTIVES + SCOPE
2
PRELIM. PROBLEM DEFINITION
PROBLEM DEFIN.
3
FEASIBILITY STUDY
NON-FEAS.
FEASIBLE
4
FINALIZE SCOPE OBJECTIVES
USER-RQMTS
FINAL SCOPE & OBJECTIVES
5
FUNCTION. ANALYSIS (PHYSICAL)
PHYSICAL-PROBLEM-DEFINITION
6
DEVELOP LOGICAL DESCRIP.
OLD-LOGICAL-DESCRIPTION
7
DEVELOP NEW LOGICAL DESCRIP.
NEW-LOGICAL-DESCRIPTION
8
FORMULATE FUNCTION. DESCRIP.
FUNCTIONAL-DESCRIPTION
9
DESIGN ANALYSIS
MODEL
10
DESIGN
NEW DESIGN
11
PROGRAMMING
PGMING-PACKAGE
12
CODING
FINAL-PROGRAM-PACKAGE
13
INSTALL
14
MAINTAIN

ANALYSIS GOAL

PROVIDE A FUNCTIONAL DESCRIPTION THAT IS:

- COMPLETE: COVERS REQUIRED AREA(S)
- UNDERSTANDABLE: NOT OPEN TO INTERPRETATION
- PARTITIONED: AID IN COMPREHENSION
- MAINTAINABLE: PORTIONS CAN BE REWRITTEN
- COMMUNICABLE: ALL RESPONSIBLE PARTIES CAN CONCUR
- USABLE: SYSTEM DESIGNERS RECOGNIZE WHAT IS TO BE DONE
- DEFINITIVE: DESCRIBES <u>WHAT</u> FUNCTIONS ARE
- ABSTRACT: SHOWS <u>WHAT</u> TO BE DONE; <u>NOT HOW</u>

GENERAL ANALYSIS

DEFINITION: STUDY OF A PROBLEM AREA PRIOR TO TAKING SOME ACTION

ACTION USUALLY MEANING SPECIFYING THE REQUIREMENTS FOR A NEW SYSTEM

COMPLETION INVOLVES:

SELECTING A TARGET AREA

DEVELOPING A FUNCTIONAL DESCRIPTION

PREDICT COST, BENEFIT, SCHEDULE, PERFORMANCE REQUIREMENTS

GETTING CUSTOMER APPROVAL

END PRODUCT OF ANALYSIS

CLASSICAL	STRUCTURED
A VERY HEAVY TOME	A VERY HEAVY TOME
LENGTHY PREPARATION	LENGTHY PREPARATION
COMPLEX	COMPLEX
NON-PARTITIONED	PARTITIONED
SELDOM USED	MADE TO BE USED
NOT SPECIFIC	SPECIFIC
CONFUSED USER	ENLIGHTENED USER
NARRATIVE	GRAPHIC + STRUCT. NARRATIVE

STRUCTURED ANALYSIS DEFINITIONS

DEFINITION: USE OF FOLLOWING TOOLS:

DATA FLOW DIAGRAMS

DATA DICTIONARY

STRUCTURED ENGLISH

DECISION TABLES
DECISION TREES (WORKING DEFINITION BY DeMARCO)

DEFINITION: ? ? ?

TWO OTHER AUTHORS DO NOT DEFINE

DEFINITION: A STUDY PROCESS USING GRAPHIC TOOLS TO PROVIDE A TARGET DOCUMENT RIGOROUSLY DEFINING CUSTOMER REQUIREMENTS IN A DEFINITIVE, PARTITIONED, NON-REDUNDANT FASHION

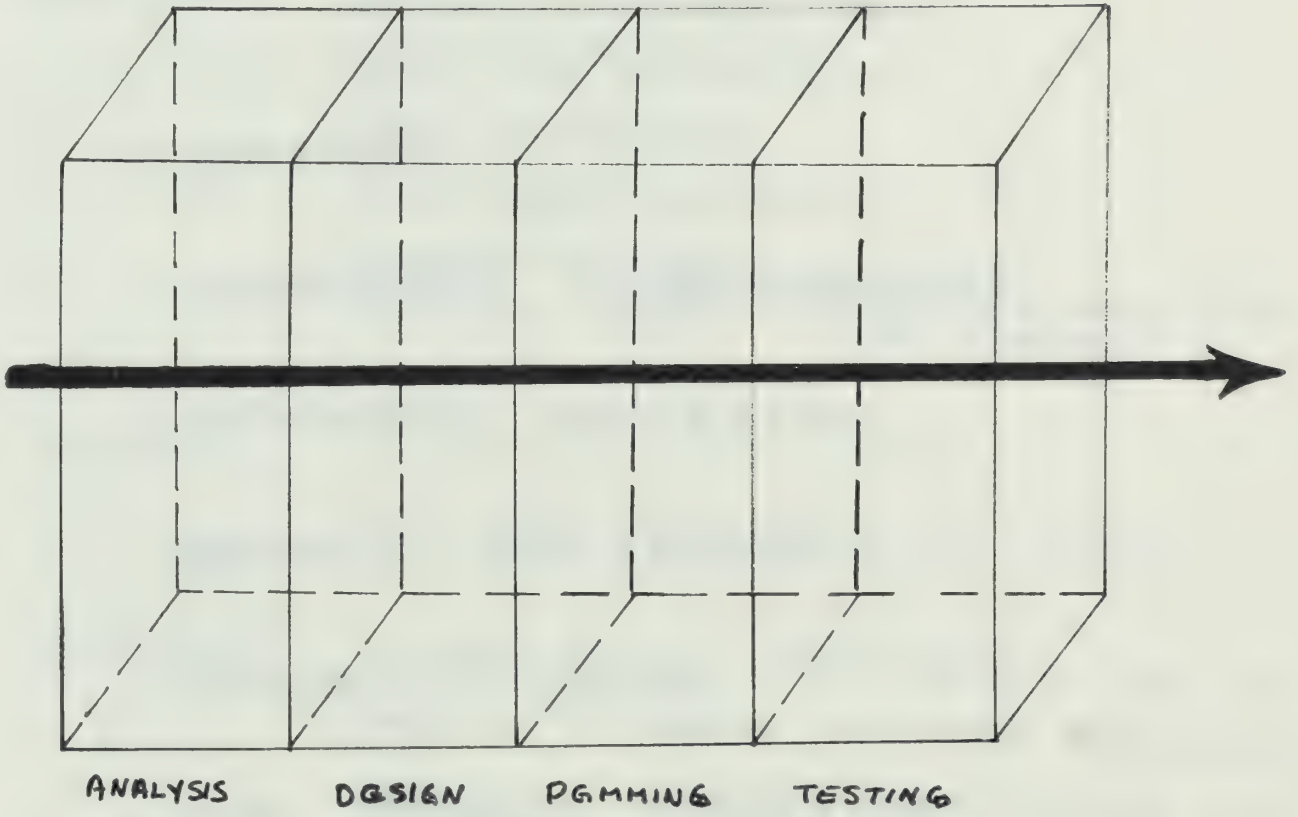
STRUCTURED ANALYSIS

NOT:

- COST BENEFIT ANALYSIS
- FEASIBILITY STUDY
- PROJECT MANAGEMENT
- EQUIPMENT SELECTION
- PERSONNEL CONSIDERATIONS
- POLITICAL

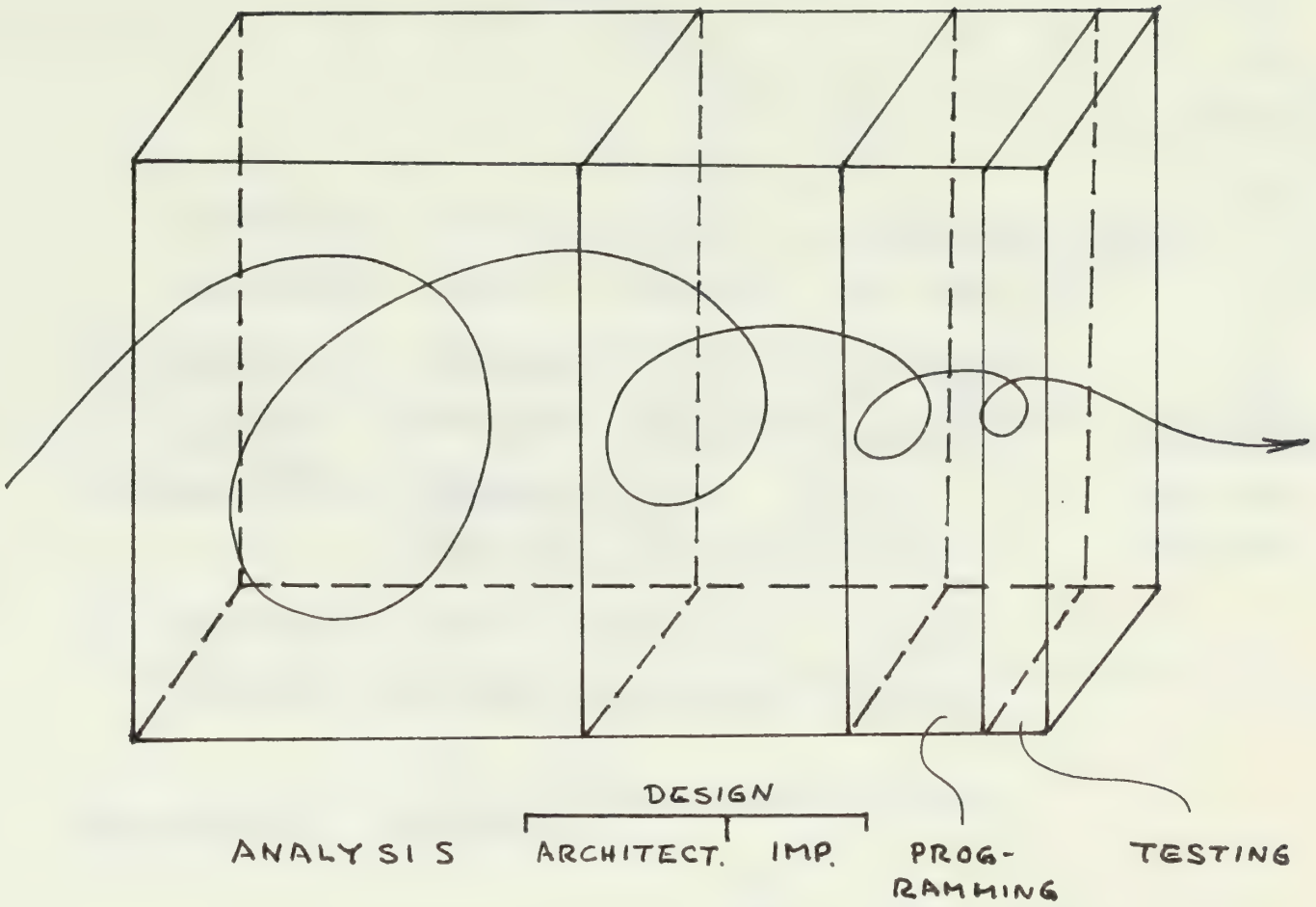
SYSTEM DEVELOPMENT

STRAIGHT LINE APPROACH



SYSTEM DEVELOPMENT

A PROCESS



5458

APPROACHES TO SYSTEMS ANALYSIS

CLASSICAL APPROACH :

STUDY PRESENT PHYSICAL SYSTEM

DEVISE A NEW PHYSICAL SYSTEM

STRUCTURED APPROACH :

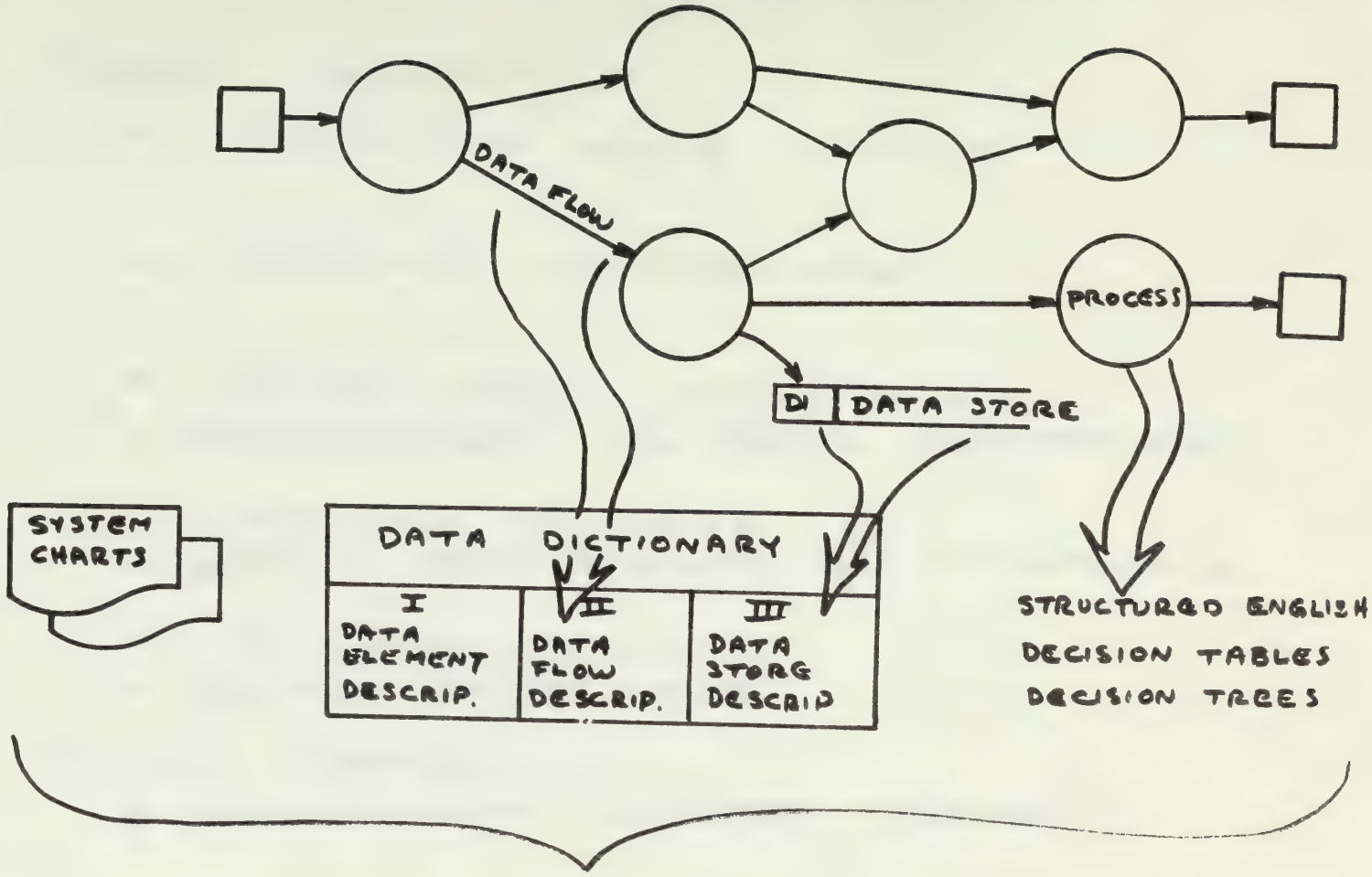
STUDY PRESENT PHYSICAL SYSTEM - DOCUMENT

DEVELOP PRESENT LOGICAL SYSTEM

DEVELOP NEW LOGICAL SYSTEM

DEVELOP NEW PHYSICAL SYSTEM

STRUCTURED SYSTEMS ANALYSIS TOOLS



NEED FOR FUNCTIONAL DESCRIPTION

- DESCRIBE PROBLEM
- COMMUNICATION BETWEEN ANALYSTS
- SIMPLIFY DESIGN TASK
- PROVIDE DOCUMENTATION FOR MAINTENANCE
- DOCUMENTATION OF DESIGN REQUIREMENTS

PRECISE FUNCTIONAL DESCRIPTION

REDUCES POSSIBILITY OF:

- SOLVING THE WRONG PROBLEM
- DESIGNER INTERPRETATION
- INCOMPATIBLE APPLICATIONS
- INTERFACE PROBLEMS
- MISINTEPRETATIONS

ALLOWS REFINEMENTS

WORK BREAKDOWN

- CONTEXT CHART
- GENERAL SYSTEM CHART
- APPLICATION CHART
- FUNCTION CHARTS
- FUNCTION DESCRIPTIONS
  - STRUCTURED ENGLISH
  - DECISION TABLE / TREE
  - WARNIER DIAGRAM
  - ⋮
- DATA DICTIONARY

DEVELOP OBJECTIVES

PRODUCT OBJECTIVES: FROM THE USER POINT OF VIEW

- SPECIFIC USER FUNCTIONS NEEDED
- ADDED NEW FUNCTIONS
- SECURITY OF DATA
- PERFORMANCE GOALS
  - RESPONSE TIME
  - RESOURCE USAGE
  - REQUIRED PERFORMANCE

PROJECT OBJECTIVES: FROM THE SYSTEM POINT OF VIEW

- PERT (GANTT CHART)
- TESTING CRITERIA

STRUCTURED ANALYSIS TOOLS

- DATA FLOW GRAPHS
- DATA DICTIONARY
- STRUCTURED ENGLISH

DATA FLOW GRAPHS

ALSO CALLED :

- DFD
- DATA FLOW DIAGRAMS
- BUBBLE CHARTS
- FLOW GRAPHS
- PROGRAM GRAPHS
- PETRI CHARTS

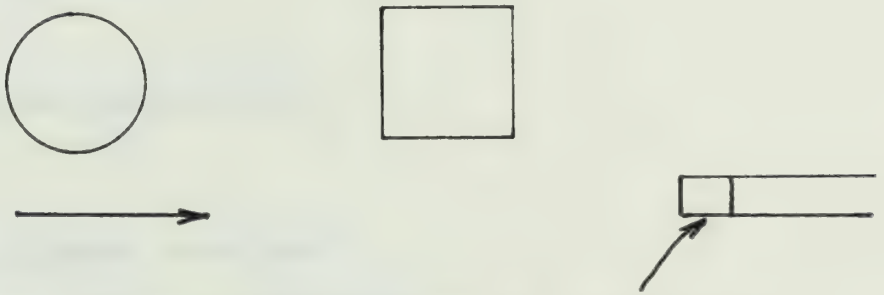
CHARACTERISTICS –

- GRAPHIC
- PARTITIONED
- EMPHASIZE DATA FLOW
- DE-EMPHASIZE CONTROL

DATA FLOW GRAPHS

A GRAPHIC REPRESENTATION OF PROCESSES, INTERFACES AND INTERRELATIONSHIPS DESIGNED TO PORTRAY SOME FUNCTION

CONSISTS OF:



MAY BE USED TO DESCRIBE THE MODEL OF THE SYSTEM THAT COULD BE:

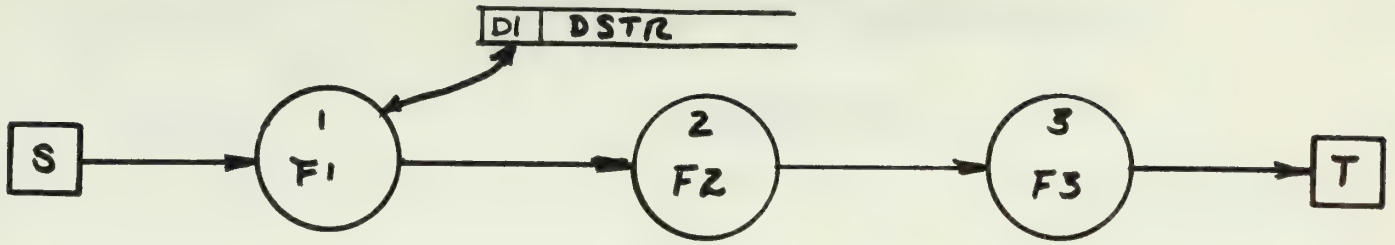
MANUAL

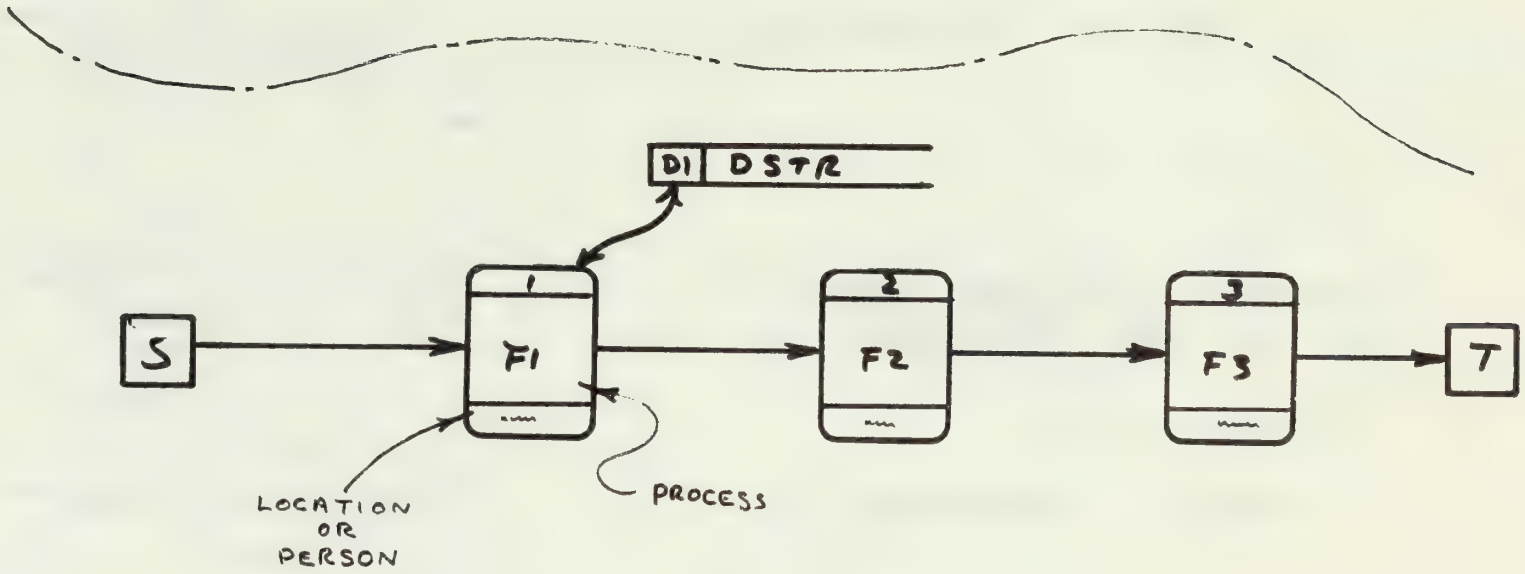
AUTOMATED

MIXED

5-16

DATA FLOW GRAPHS

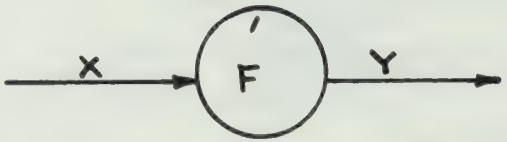




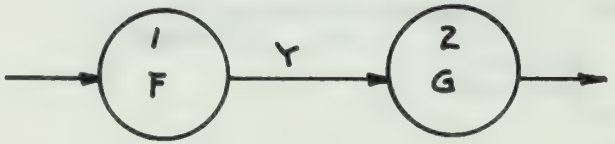
CONVENTION ILLUSTRATION

DATA FLOW GRAPHS

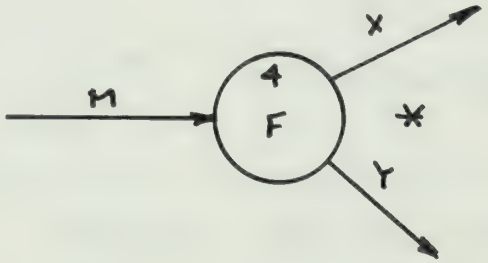
BASIC ELEMENTS



F TRANSFORMS X INTO Y

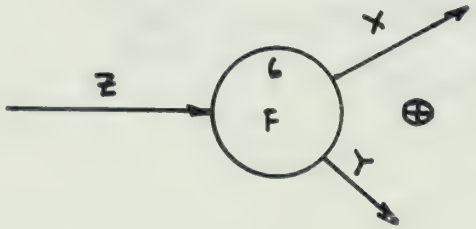


Y PRODUCED BY F IS USED BY G



F PRODUCES X AND Y

(CONJUNCTION)

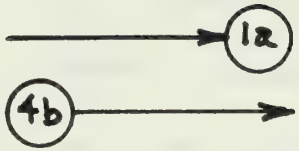


F PRODUCES X OR Y

(DISJUNCTION)

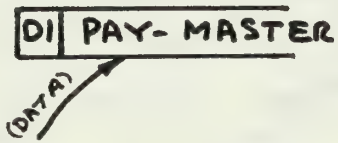
DATA FLOW GRAPHS

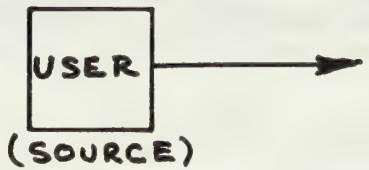
ADDITIONAL NOTATION



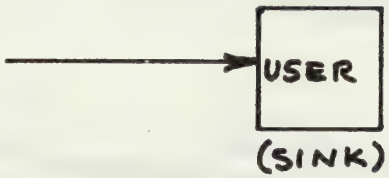
CONNECTOR

DATA STORE ACCESSED

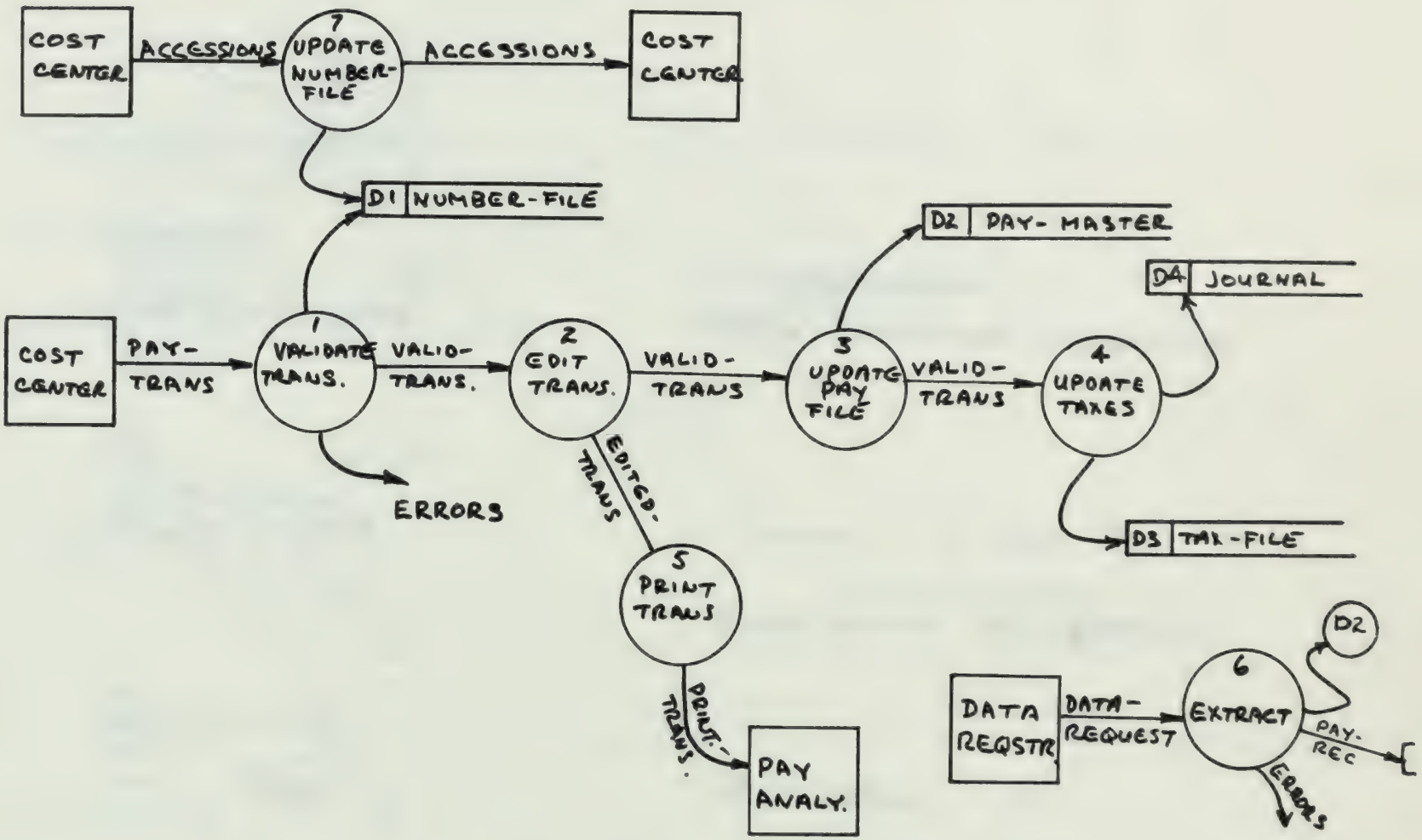




MAN-MACHINE INTERFACE BOUNDARY



DATA FLOW GRAPH EXAMPLE



DATA FLOW GRAPH CONVENTIONS

1. START WITH INPUTS & WORK TOWARD CENTRAL TRANSFORM AND FINAL OUTPUT. (OR OUTPUTS BACKWARDS) (INSIDE OUT).
2. EACH BUBBLE REPRESENTS A TRANSFORM (CHANGE IN DATA)
3. DO NOT SHOW CONTROL
4. SHOW ONLY SIMPLE ERROR PATHS
5. IGNORE INITIALIZATION & TERMINATION - ASSUME PERPETUAL MOTION
6. LABEL SPECIFICALLY - DATA, DATA STORES, TRANSFORMS
7. USE CONJUNCTION, DISJUNCTION SYMBOLS (* ⊕) SPARINGLY
8. INTRODUCE DATA ONLY WHEN NEEDED
9. ASSURE CORRECTNESS AT EVERY LEVEL OF DETAIL
10. DONT FLOW CHART

DATA FLOW GRAPHS

PARTITIONING : BREAKING A SYSTEM INTO ITS COMPONENT PARTS

RESULT : LEVELS OF DFG'S

PARTITIONING CONTINUES BREAKING DOWN UNTIL

- BOTTOM LEVEL "REASONABLY" PRIMITIVE

OR

- BOTTOM LEVEL SHOWS ONE PAGE OF NARRATIVE PER BUBBLE

OR

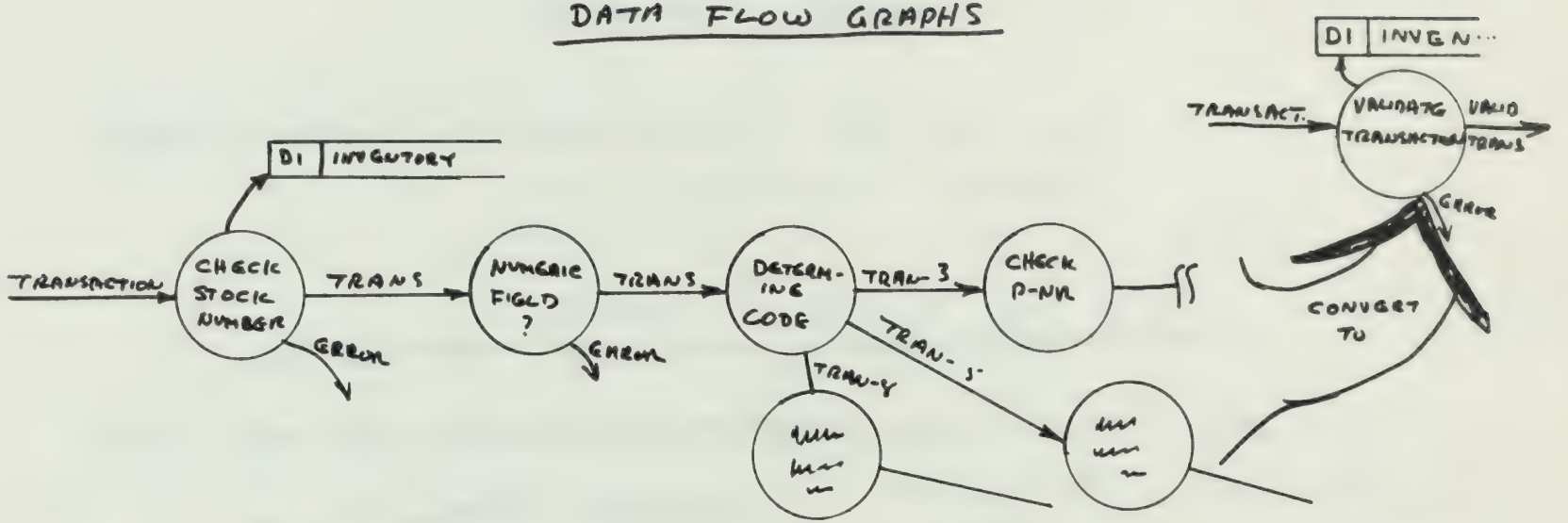
- BOTTOM LEVEL HAS 7±2 BUBBLES AND CANNOT LOGICALLY BE BROKEN INTO MORE BUBBLES

DATA FLOW GRAPHS

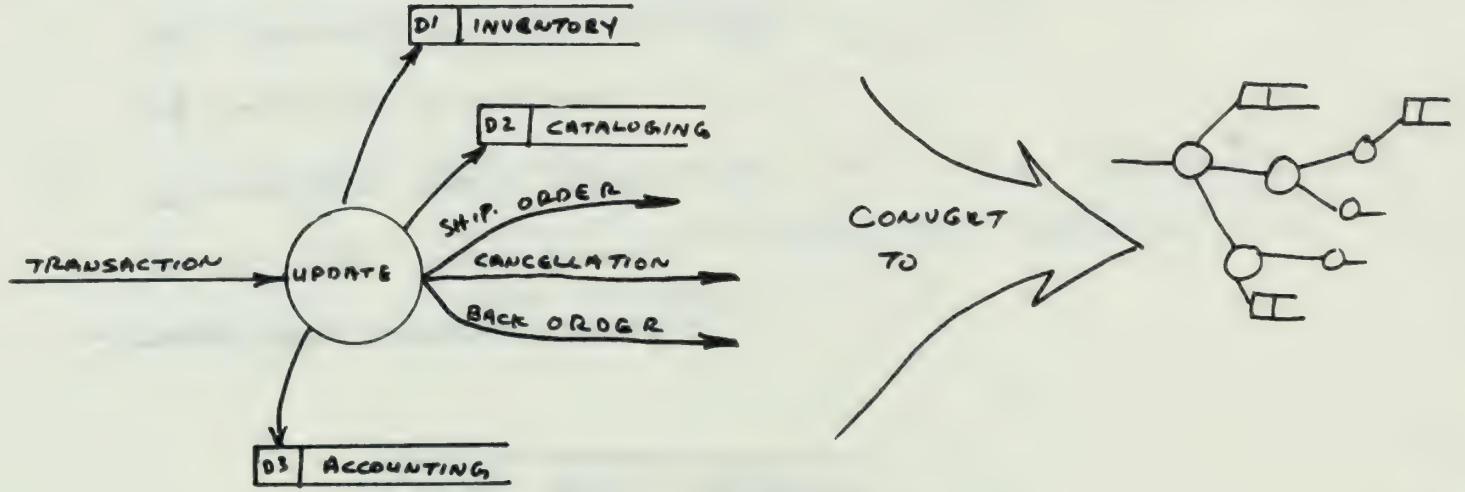
— TESTS FOR CORRECTNESS :

- MISSING DATA FLOWS
- USELESS DATA FLOWS
- MISSING FUNCTIONS
- INCORRECT PARTITIONING
- UNCLEAR LABELS (DATA FLOWS, FUNCTIONS, DATA STORES)
- CONTROLS PRESENT
- CONCEPTUALLY INCORRECT
- UNKNOWN DATA FLOWS / FUNCTIONS / DATA STORES

DATA FLOW GRAPHS



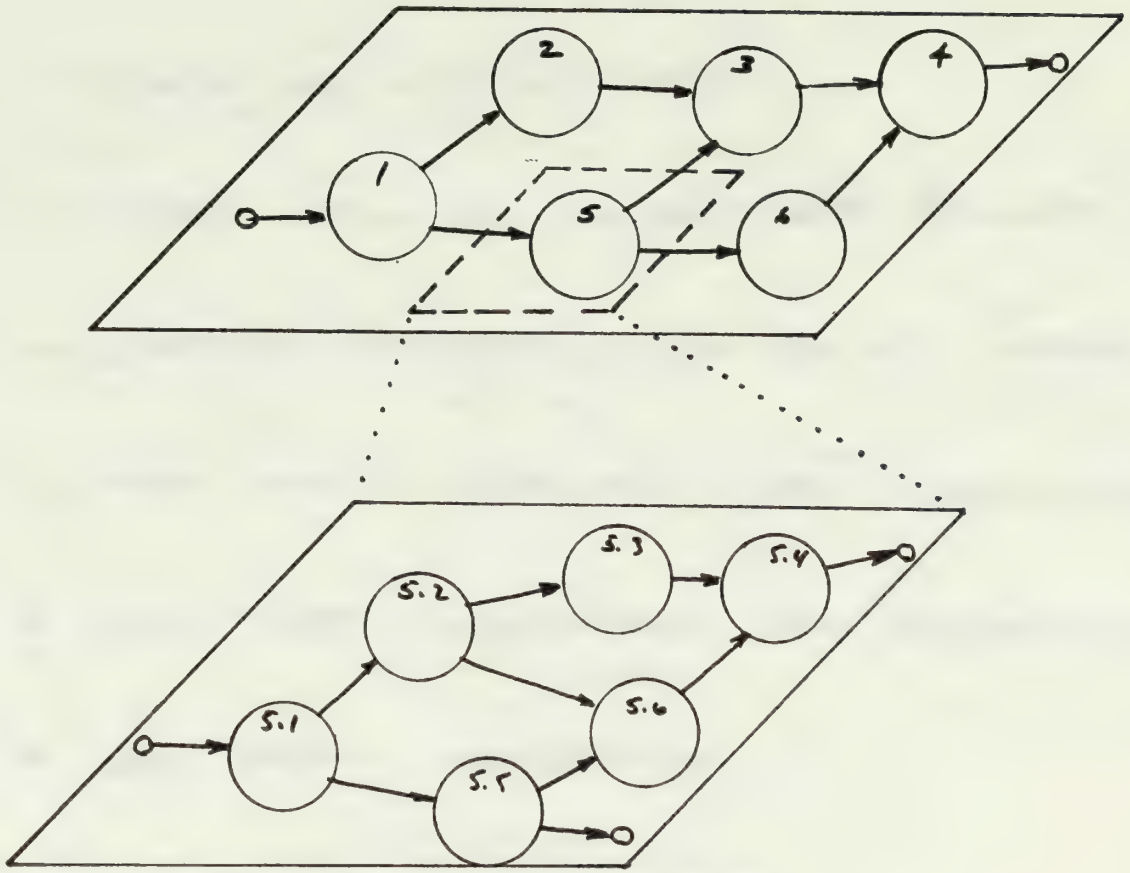
TOO DETAILED



TOO GROSS

DATA FLOW GRAPHS

LEVELS



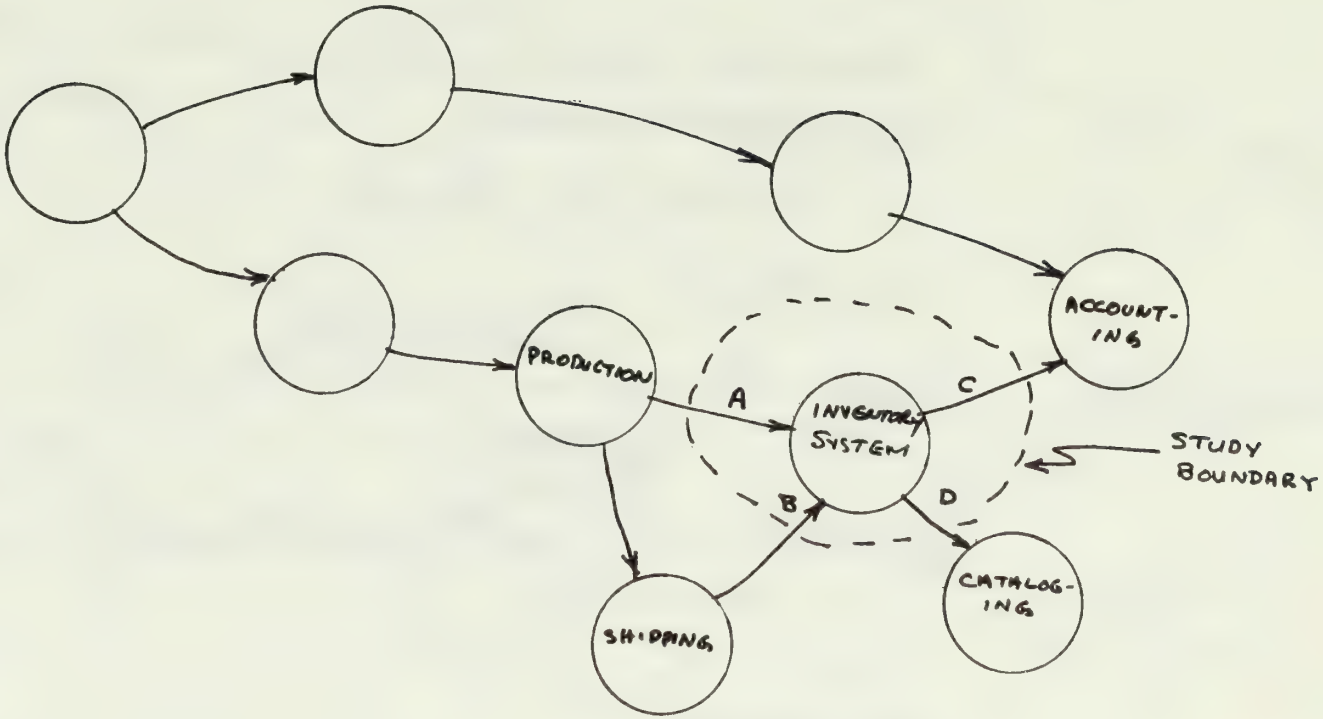
5-124

DATA FLOW GRAPHS

ADVANTAGES OF LEVELED DFGS:

- TOP DOWN APPROACH
- TOP LEVEL FUNCTIONS FOR MANAGERS
- LOW LEVEL FUNCTIONS FOR OPERATIONAL USERS
- FORCES CONSIDERATION OF ALL ASPECTS
- USES NON-ADP TERMS FOR BETTER COMMUNICATION
- EASE OF MAINTENANCE

DATA FLOW GRAPHS

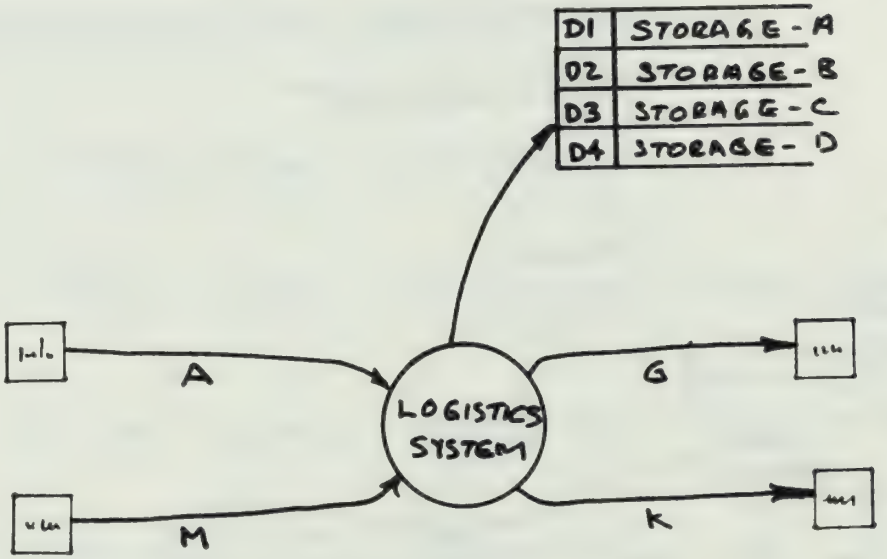


PARTITIONING

(TOP DOWN ANALYSIS)

DATA FLOW GRAPH

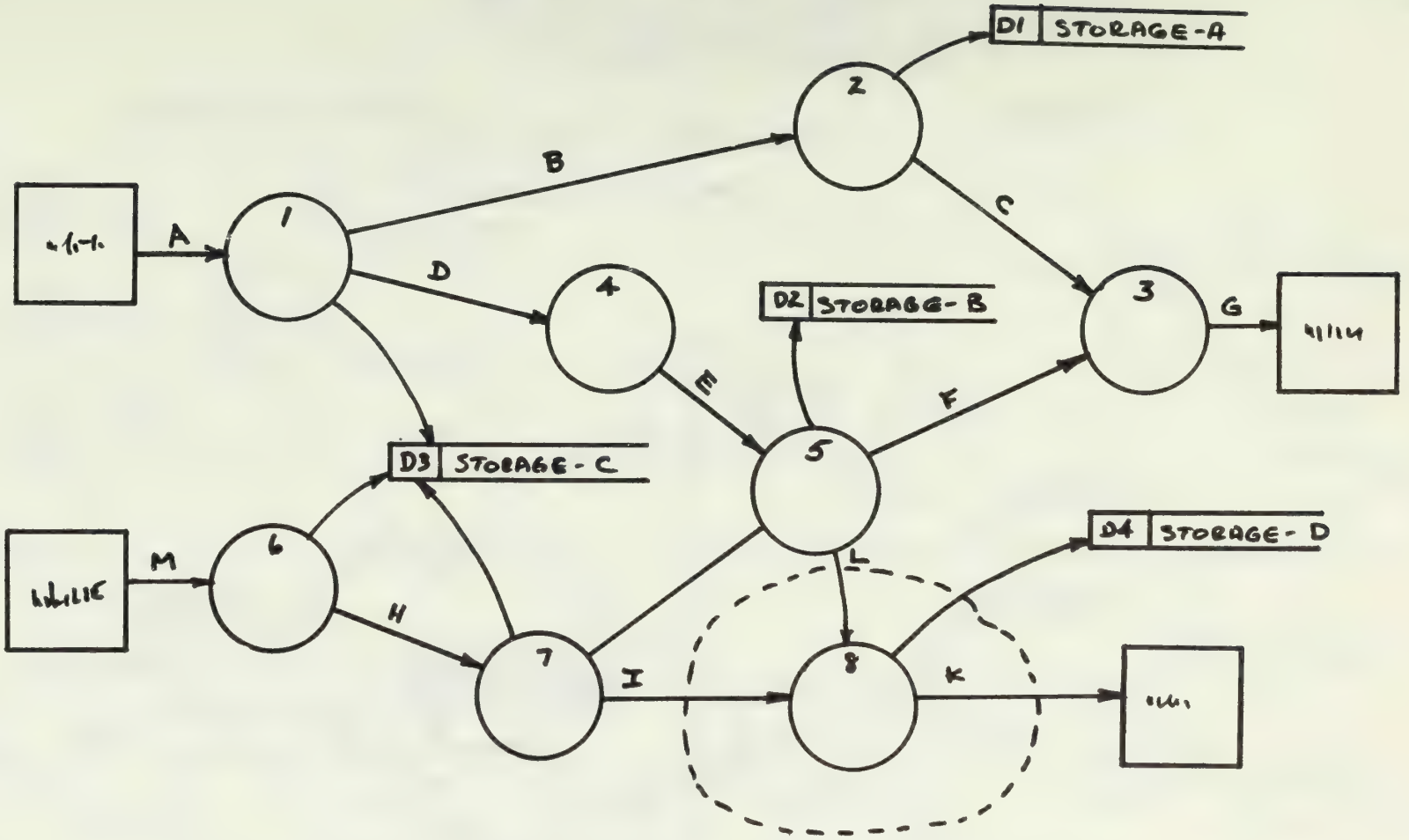
CONTEXT DIAGRAM



DOMAIN OF STUDY

PARTITIONING
(TOP DOWN ANALYSIS)

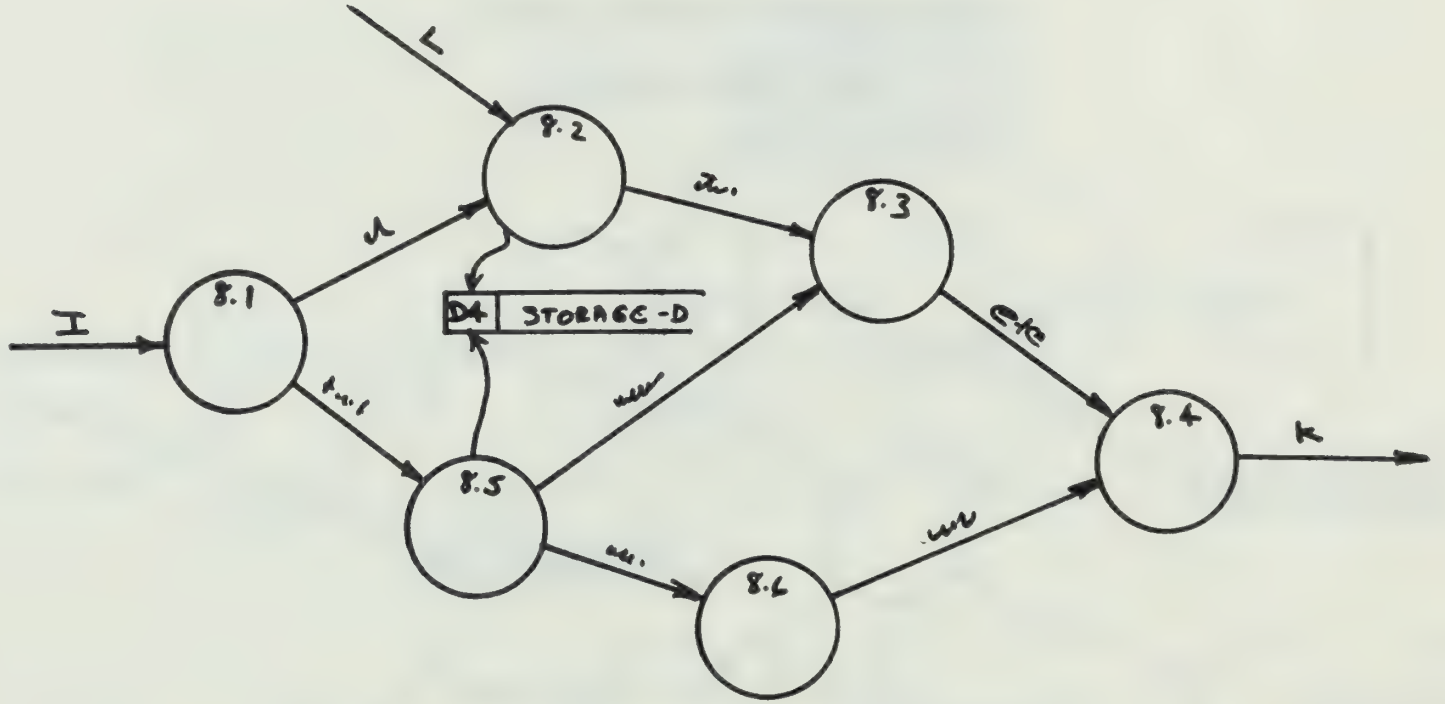
DATA FLOW GRAPH



PARTITIONING

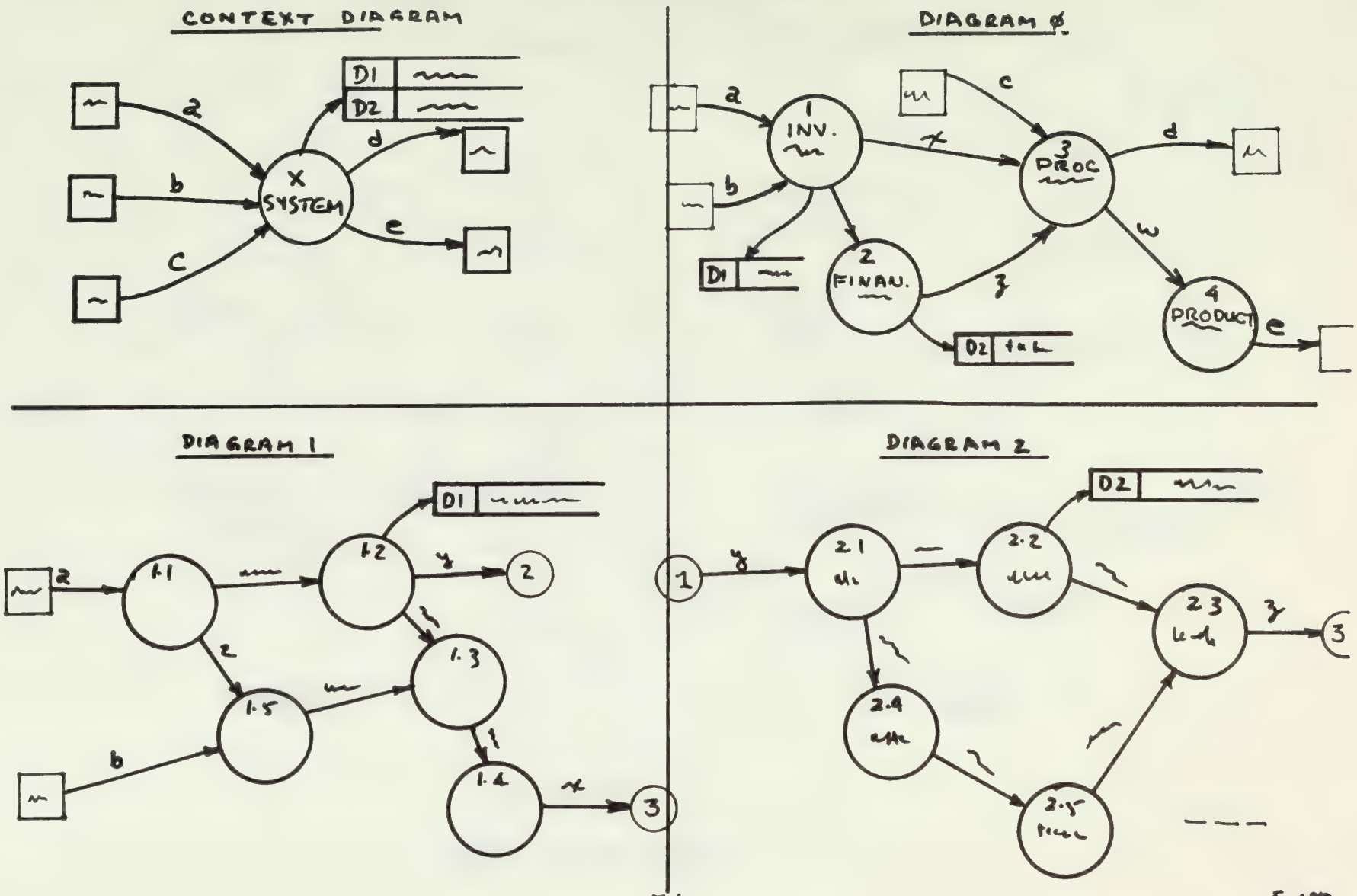
(TOP DOWN ANALYSIS)

DATA FLOW GRAPH



PARTITIONING
(TOP DOWN ANALYSIS)

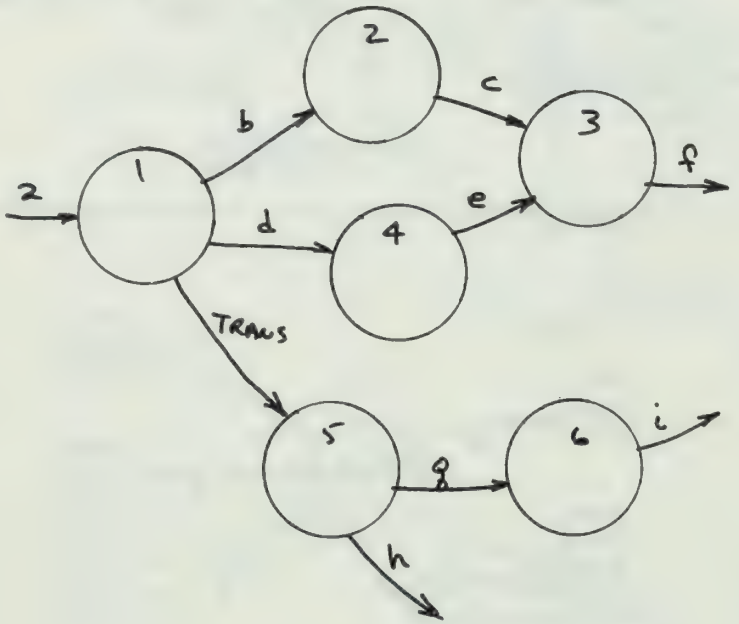
DATA FLOW GRAPH
LEVEL CONVENTIONS



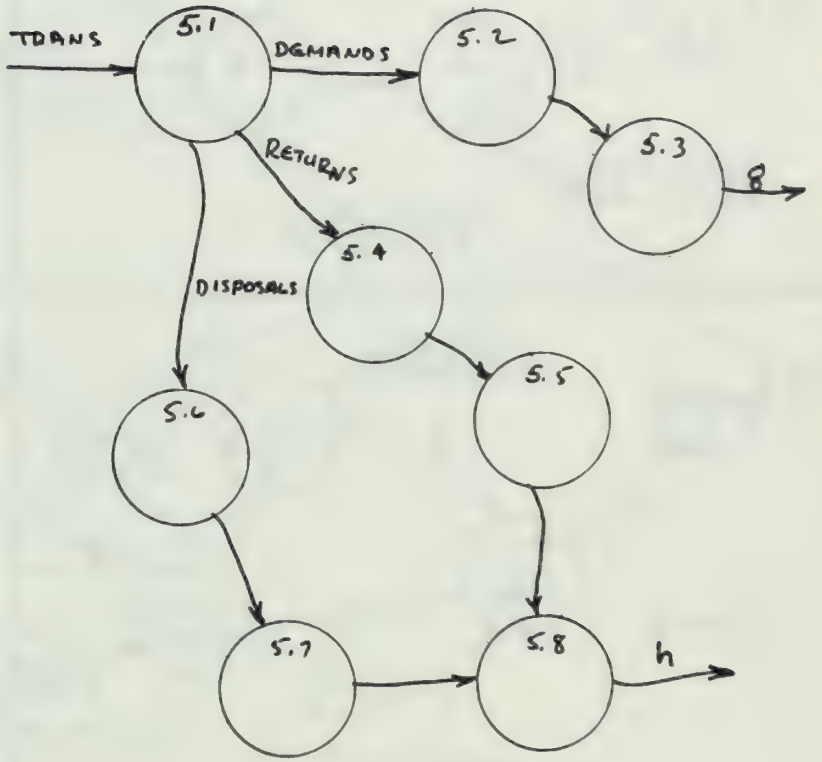
DATA FLOW GRAPH

BALANCING

PARENT



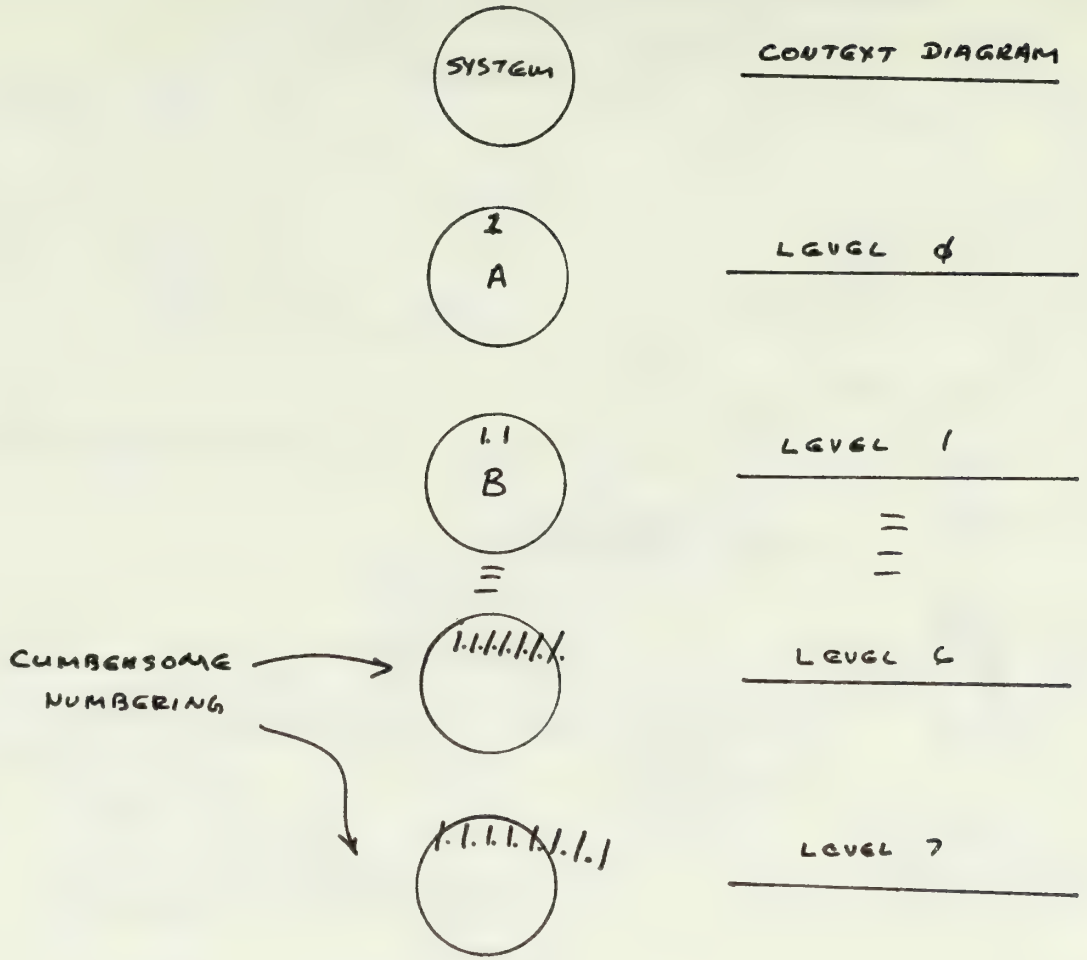
CHILD



DATA FLOW DIAGRAMS

NUMBERING CONVENTIONS

PROBLEMS

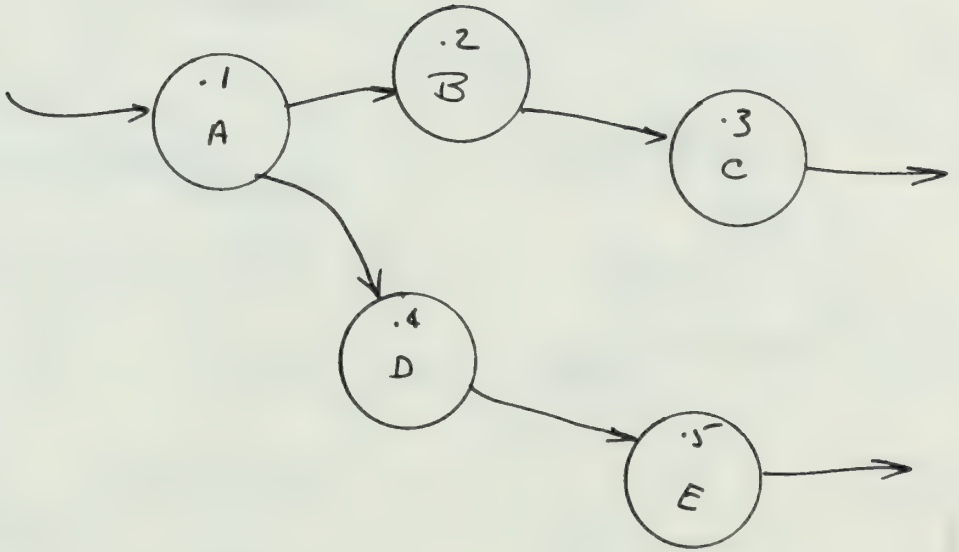


5-182

DATA FLOW GRAPH

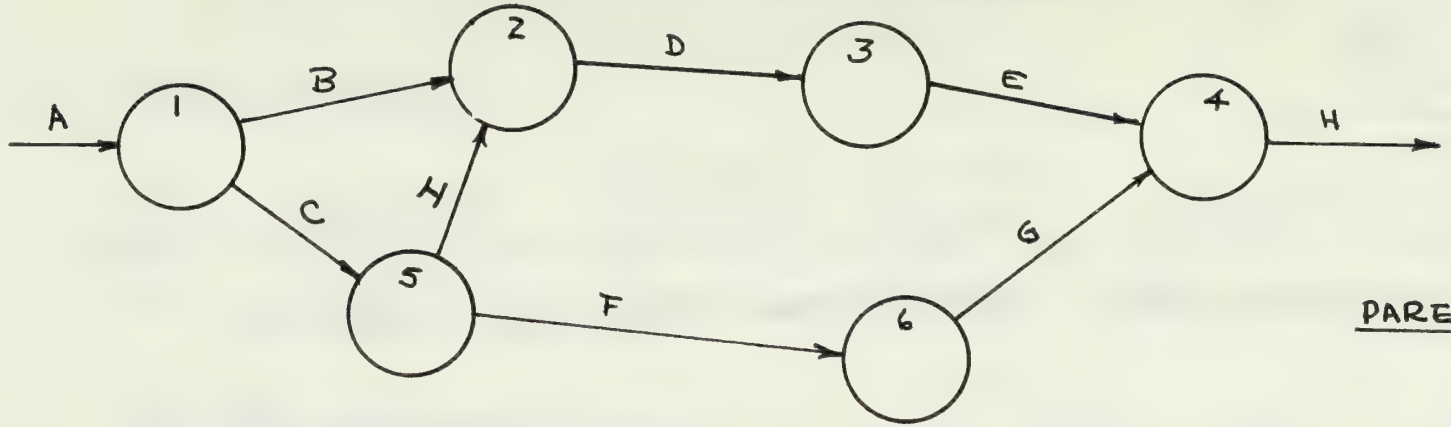
NUMBERS REDUCTION

DIAGRAM 1.1.1.1.1.1

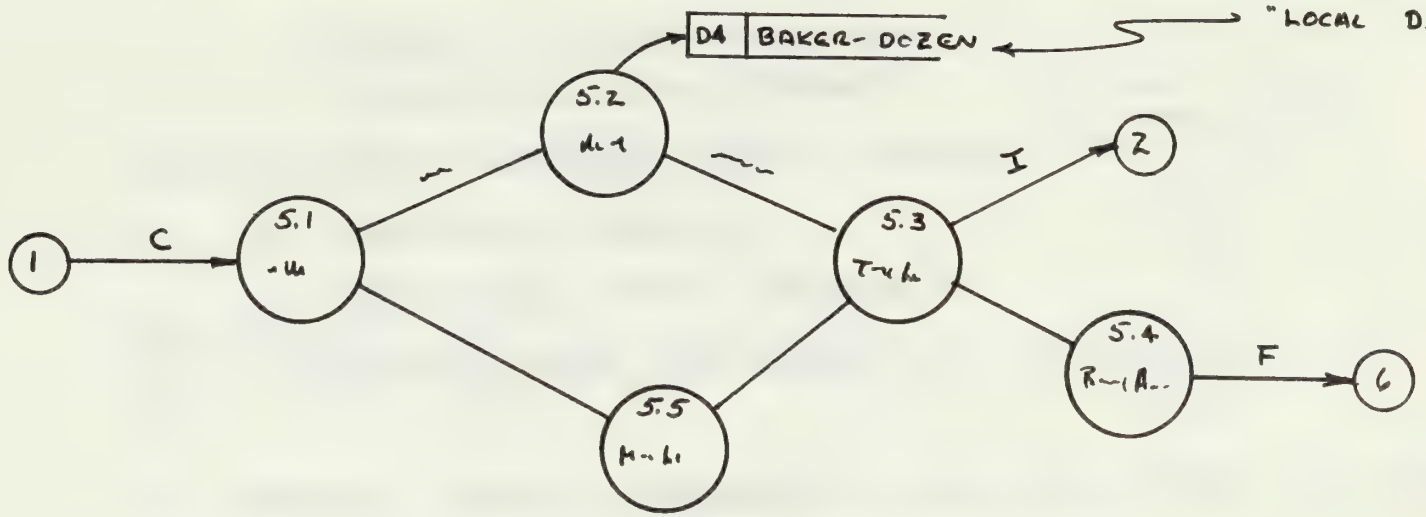


DATA FLOW GRAPH

LOCAL DATA STORES



PARENT



CHILE

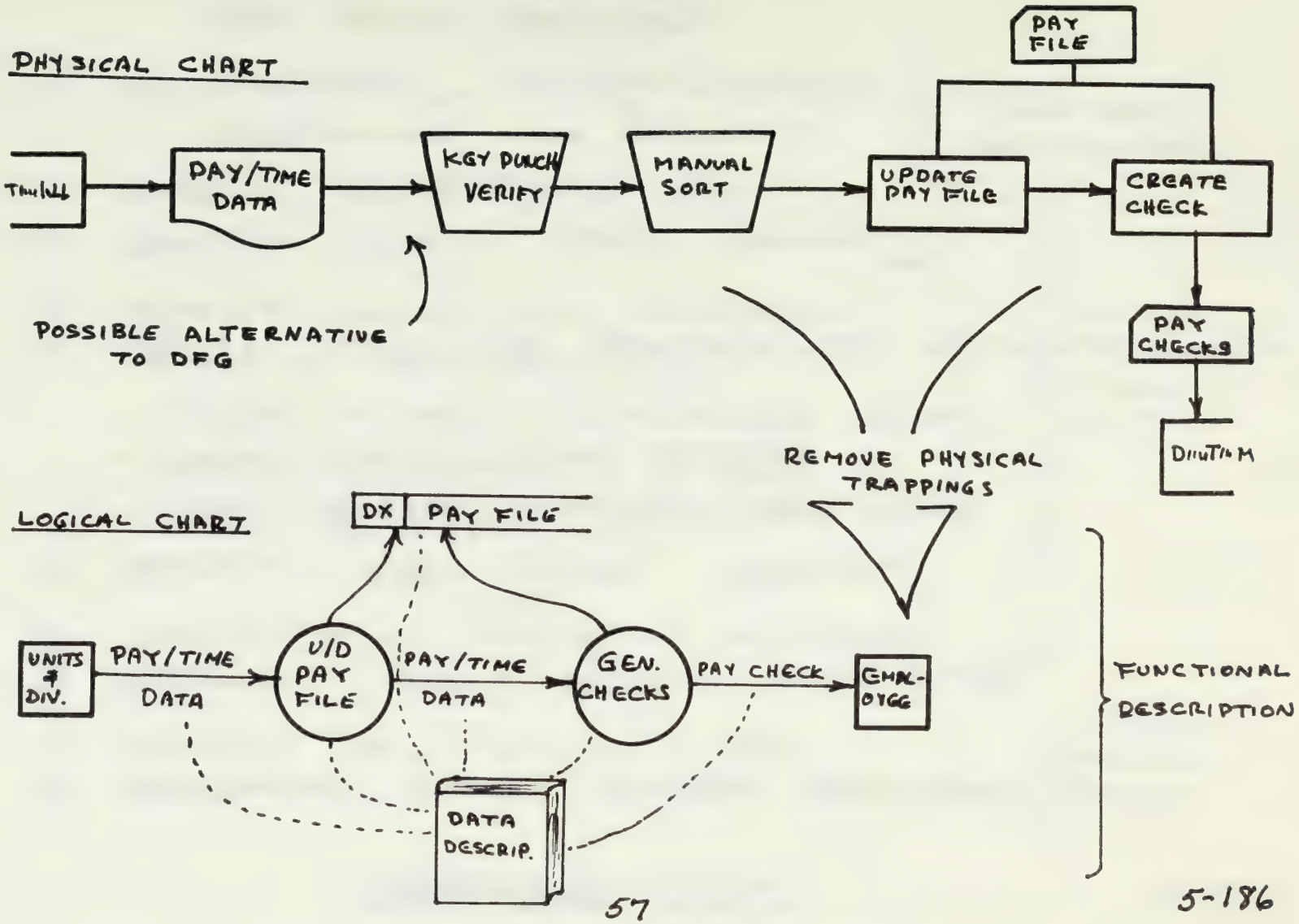
5-184

DATA FLOW GRAPHS

- LOWEST LEVEL BUBBLES MUST BE DOCUMENTED:
  - STRUCTURED ENGLISH
  - DECISION LOGIC TABLES
  - DECISION TREES
  - STRUCTURED FLOW CHARTS
  - WARNIER DIAGRAMS
  - NASSI- SCHNEIDERMAN
- GENERAL RULE:
  - ONE PROCESS DESCRIPTION PER BUBBLE

ANALYSIS

PHYSICAL VS LOGICAL



DEVELOPING DFG'S

1. DETERMINE SCOPE, SYSTEM BOUNDARY

2. IDENTIFY INPUTS, OUTPUTS, MAJOR FILES

3. DEVELOP FIRST DRAFT (PHYSICAL)
   - SHOW PRESENT DOCUMENT/ DATA FLOW
   - SHOW PRESENT DATA STORAGE

4. REVIEW 2 & 3 FOR COMPLETENESS, ACCURACY

5. DEVELOP SECOND DRAFT (PHYSICAL)
   IMPROVE COMPREHENSION BY:
   - DUPLICATING DATA STORAGES
   - DUPLICATING EXTERNAL ENTITIES
   - ADD NOTES / COMMENTS

   } WALK THROUGH WITH USER

5-187A

6. DEVELOP LOGICAL DFG - OLD
   REMOVE PHYSICAL TRAPPINGS

7. WALK THROUGH WITH USER; ANNOTATE CHANGES

8. DEVELOP LOGICAL DFG - NEW

9. DEVELOP LOWER LEVEL PROCESSES
   - DOWN TO PRIMITIVE LEVEL
   - REVIEW TO ASSURE COMPLETENESS, ACCURACY
   - WATCH CONSERVATION OF DATA
   - KEEP "LUMP LAW" IN MIND

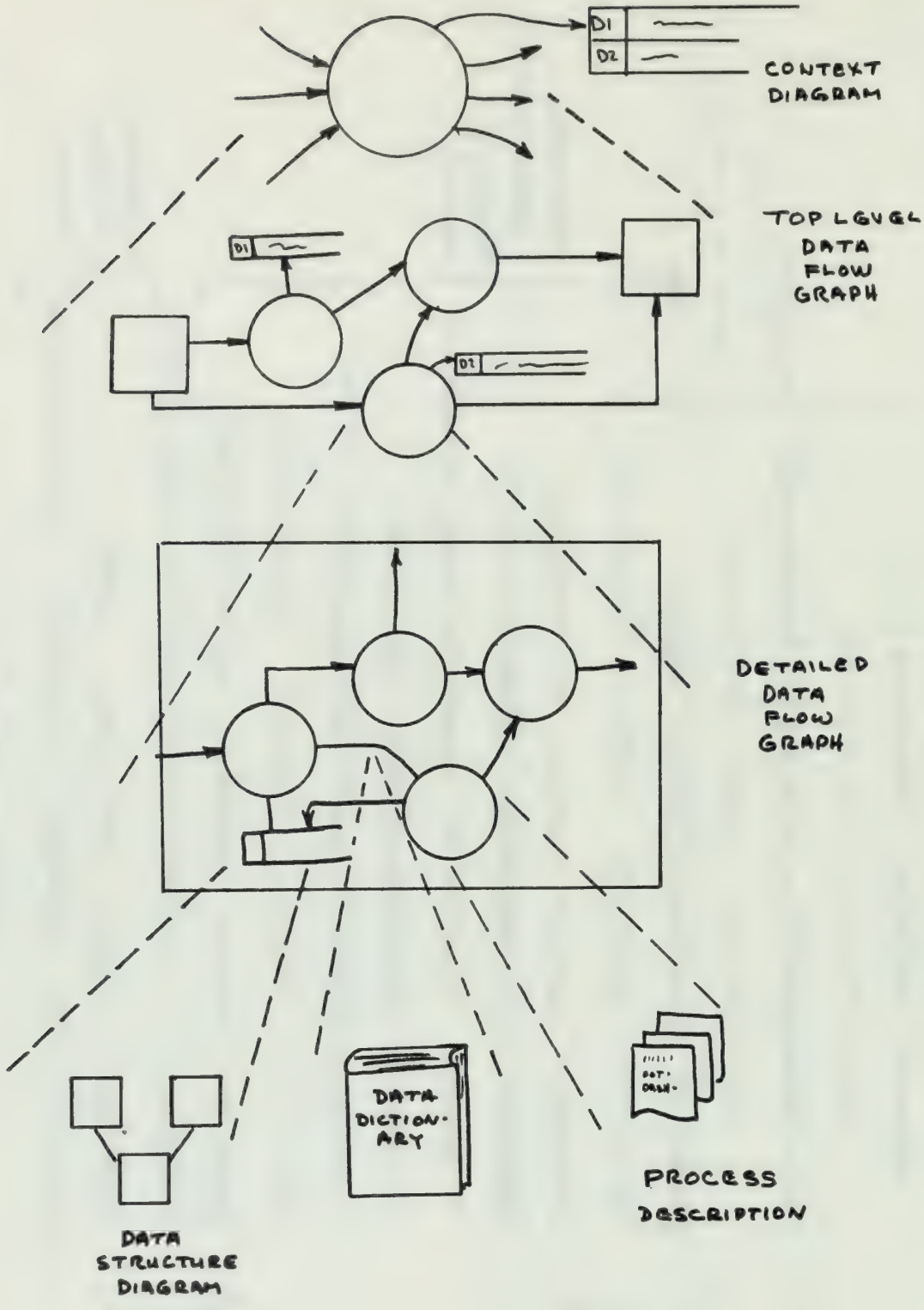
10. DEVELOP FINAL DRAFT AS NECESSARY

INCLUDE NEW LOGICAL REQUIREMENTS

11. DEVELOP NEW PHYSICAL DFG

INCLUDE PHYSICAL RQMTS

5-187B


D1
D2
CONTEXT DIAGRAM
TOP LEVEL DATA FLOW GRAPH
DETAILED DATA FLOW GRAPH
DFG LEVELS
DATA DICTION-ARY
DATA STRUCTURE DIAGRAM
PROCESS DESCRIPTION

DATA DICTIONARY

- PURPOSE -

DEFINE SYSTEM DATA

- DATA FLOWS
- DATA FLOW COMPONENTS (DATA ELEMENTS)
- DATA STORES

- REQUIREMENTS -

- DEFINITIONS EASILY ACCESSIBLE BY NAME
- MINIMUM REDUNDANCY
- SIMPLE TO UPDATE
- CONVENTIONS FOR WRITING SHOULD BE SIMPLE

DATA DICTIONARY

- CONVENTIONS -

=	IS EQUIVALENT TO
+	AND
[ ]	EITHER - OR
$\overset{N}{\{ \}}$	REPETITION
( )	OPTIONAL
*	COMMENTS

DATA DICTIONARY

- CONVENTION ILLUSTRATIONS -

INVENTORY-RECORD = STOCK-NUMBER + QUANTITY + ····

STOCKAGE-POINT = [DEPOT / SUPPLY-SHIP]

STORAGE-LOCATION = $_{1}^{41}\{$ BIN $\}$ OR $1\{$ BIN $\}_{41}$

PAYMENT-AMOUNT = [DOLLAR-AMOUNT / NON-DOLLAR-AMOUNT *YEN ONLY*]

EMPLOYEE-RECORD = NAME + SOC-SEC-NR + (SEX) +
DATE-OF-RECORD

DATA DICTIONARY

VALID-TRANSACTION = * OUTPUT OF 4.1.6 *

= STOCK-NR + PART-NR +
DEPOT + QUANTITY + DATE

DEPOT-DATA = * COMPONENT OF STOCK-ITEM *

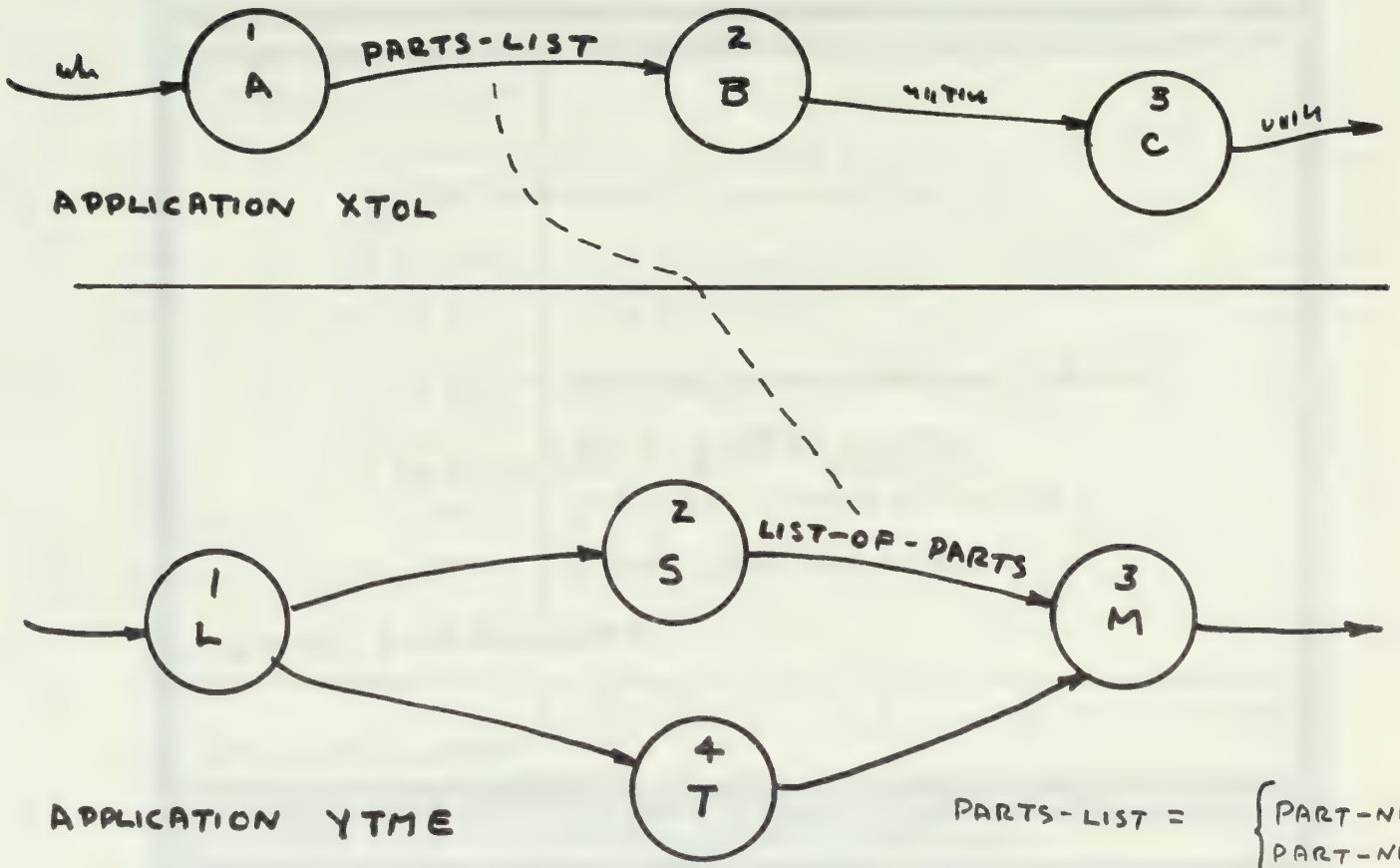
$= \overset{8}{\underset{1}{[}} \{ \text{DEPOT-NAME} + \text{ACCOUNT} + \text{QTY} \}$

ITEM-LINE = QUANTITY + IDENT + PRICE +
TOTAL-PRICE * QUANTITY x PRICE *

DATA DICTIONARY

- ALIASES -

ANOTHER NAME FOR THE SAME THING



PARTS-LIST = { PART-NR + PART-NAME }

= LIST-OF-PARTS

LIST-OF-PARTS = PARTS-LIST

5-193

DATA DICTIONARY

DATA ELEMENT NAME: TYPE-OF-ITEM

ALIASES: TYPE-ITEM; ITEM-TYPE

COMPOSITION: A

VALUES & MEANINGS:

L	LOW DOLLAR VALUE
M	MEDIUM DOLLAR VALUE
H	HIGH DOLLAR VALUE
S	SUPER HIGH DOLLAR VALUE
P	PITTANCE

NOTES:

5-195

DATA DICTIONARY

DATAFLOW NAME: RESPONSE

ALIASES: NONE

COMPOSITION:

ITEM-STUDY-NUMBER +
DEPOT-DEMAND-SUMMARY +
$\overset{5}{\underset{1}{\{}}$ CUSTOMER-BY-QUARTER-SUMMARY $\}$ +
$\overset{6}{\underset{1}{\{}}$ ASSETS-BY-CONDITION-CODE $\}$ +
(SUBSTITUTE-ITEM-NUMBER)

NOTES: ONLY ITEM MARS GET RESPONSES
RESPONSE TIME = 30SEC MAX.

DATA DICTIONARY

DATA STORE NAME: ACCOUNTING	D3
ALIASES: NONE	
COMPOSITION: ACCOUNT-NUMBER + DATE + CUSTOMER-NAME + CUSTOMER-ADDR + AMOUNT PAID	
ORGANIZATION: SEQUENTIAL BY ACCOUNT-NUMBER	
NOTES: CHANGES TO THIS DATA MUST BE APPROVED BY THE CHIEF ACCOUNTING OFFICER	

DATA DICTIONARY

GLOSSARY ITEM: STANDARD-COMPARISON

DESCRIPTION:

RELATION BETWEEN EMPLOYEE'S SALARY AND THE SALARIES OF ALL IN SAME GRADE EXPRESSED AS A STANDARD DEVIATION.

DATA DICTIONARY

DESCRIPTION OF DATA ELEMENTS

1. ALIASES
2. RELATED DATA ELEMENTS
3. VALUES - CONTINUOUS VS DISCRETE
4. LENGTH & TYPE OF CHARACTERS
5. ENCODING
6. EDITING INFORMATION

DATA DICTIONARY

IMPLEMENTATION

1. CREATE AN ENTRY PAGE FOR EACH ITEM

2. PLACE ITEM NAME AT TOP

3. WRITE DEFINITION USING CONVENTIONS

4. PROVIDE NOTES ABOUT PHYSICAL OR OTHER ASPECTS

5. PROVIDE FOR EACH ALIAS WITH CROSS REFERENCE

6. ENTRIES IN ALPHABETICAL SEQUENCE WITHIN DATA ELEMENT, DATA FLOW, DATA STORE

DATA DICTIONARY

- MANUAL
  - SMALL PROJECTS MAY NOT REQUIRE AUTOMATED DICTIONARY
  - SOME ORGANIZATIONS DO NOT HAVE ACCESS TO COMMERCIAL DD PACKAGE

- ADMINISTRATION
  - HANDLED BY LIBRARIAN
  - KEPT ON 5X8 CARDS OR LOOSELEAF BINDER

DATA DICTIONARY

COMMERCIAL DD PACKAGES

DISADVANTAGES:

COSTLY

(SOFTWARE + USER TRAINING + OVERHEAD)

NONE IDEAL

INTERFACE TO PACKAGE IS COMPLICATED

(MAY REQUIRE ONE PERSON FULL TIME)

MORE ORIENTED TO LATER PHASES OF DEVELOPMENT THAN ANALYSIS

DATA DICTIONARY

AUTOMATED

COMMERCIAL PACKAGES AVAILABLE

- DATA DICTIONARY (IMS)
- DATA CATALOG
- DATA MANAGER
- LEXICON

ADVANTAGES:

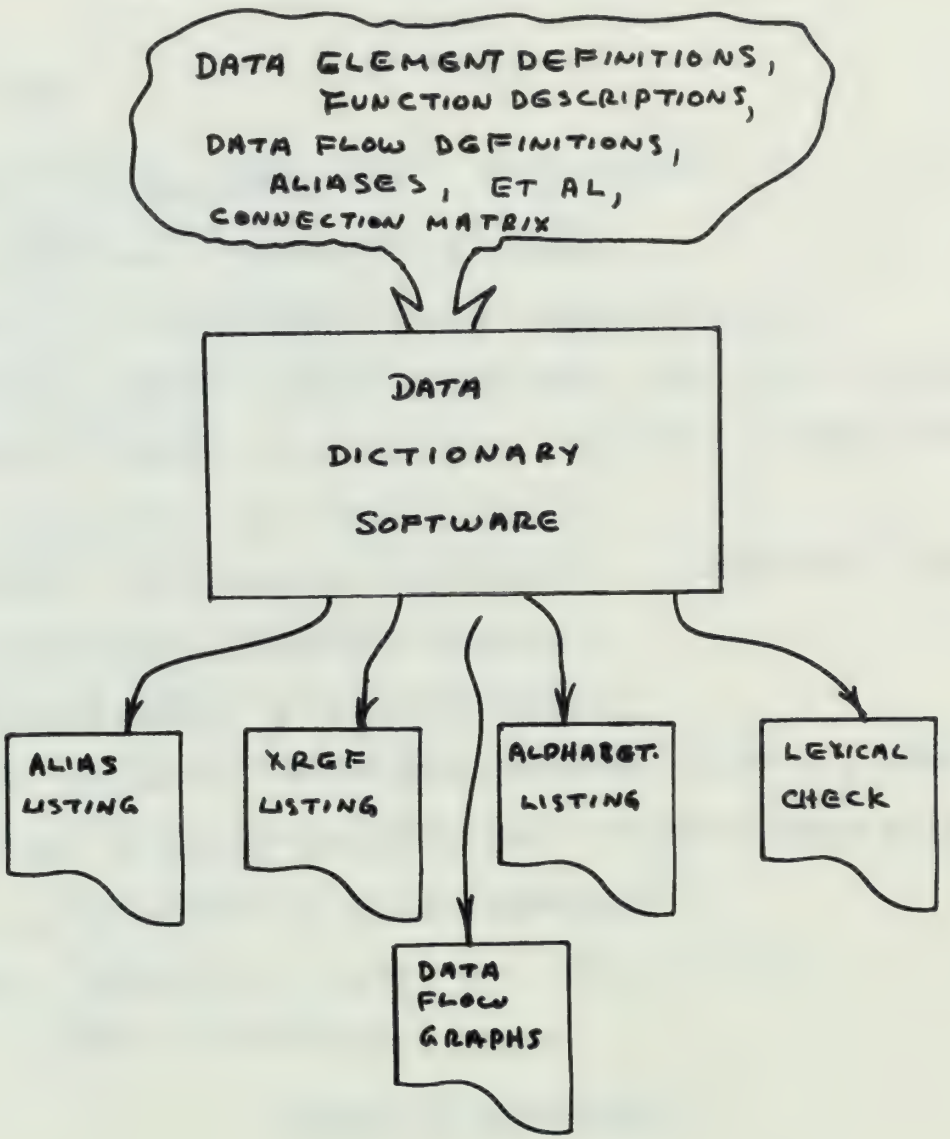
- ALIAS MANAGEMENT
- HIERARCHICAL LISTINGS
- CROSS REFERENCE LISTINGS
- CORRELATION BETWEEN DATA FLOWS AND:
  - SUPERORDINATES / SUBORDINATES
  - SOURCE
  - DESTINATION

DATA DICTIONARY

– IDEAL DD PROCESSOR –

- ACCEPT DEFINITIONS FOR ALL DATA FLOWS, DATA ELEMENTS & FUNCTIONS
- PROVIDE SIMPLE DEFINITION CONVENTIONS
- ALLOW TOTALLY NON-REDUNDANT INPUT (TRACKS SUPERORDINATES & SUBORDINATES)
- PROVIDE FOR SIMPLE UPDATES
- PROVIDE CONSISTENCY CHECKING (DUPLICATE NAMES, UNCONNECTED ALIASES, ETC.)
- PROVIDE CROSS REFERENCE LISTINGS (DATA ELEMENTS TO DATA FLOWS; DATA REQUIRED BY EACH FUNCTION ....)
- DRAW & MAINTAIN DATA FLOW GRAPHS
- INADVERTANT SOURCE & SINKS
- VERIFY BALANCING

IDEAL DD PROCESSOR



STRUCTURED ENGLISH

WHAT IS IT ?

A MEANS OF DESCRIBING IN WORDS, THE DESIGN OF A SYSTEM / PROGRAM

(NED CHAPIN)

ALIASES:

(PSEUDO CODE)

STRUCTURED NARRATIVE

STRUCTURED TEXT

PROGRAM DESIGN LANGUAGE

STRUCTURED ENGLISH

CHARACTERISTICS:

1. DESCRIBES A FUNCTION IN ORDINARY PROSE

2. INITIALLY, IMPOSES FEW VOCABULARY RESTRICTIONS

3. ASSIGNS SPECIAL SIGNIFICANCE TO A MERE EIGHT (8) WORDS:

START	IF	END-IF	LOOP
END	ELSE	EXIT	END-LOOP

4. ORGANIZES DESCRIPTION:
   - IDENTIFYING NAME
   - INPUT
   - OUTPUT
   - FUNCTIONS

STRUCTURED ENGLISH

(CHARACTERISTICS)

5. AN ITERATIVE PROCESS

6. CONSISTENT WITH STRUCTURED PROGRAMMING

7. DOES NOT REQUIRE, BUT MAY BE USED WITH GRAPHS OR CHARTS

STRUCTURED ENGLISH

ITERATIONS

1. STUDY AVAILABLE KNOWN FUNCTIONS

2. STUDY VERBS, NOUNS, SPECIAL WORDS ALREADY USED IN THE FUNCTION

3. SELECT WHERE TO NARROW VOCABULARY

4. DETERMINE WHERE TO EXPAND STATEMENTS FOR A MORE COMPLETE, SPECIFIC DESCRIPTION

5. IDENTIFY INPUTS & OUTPUTS

6. DESCRIBE FUNCTIONS

7. REVIEW FOR COMPLETENESS

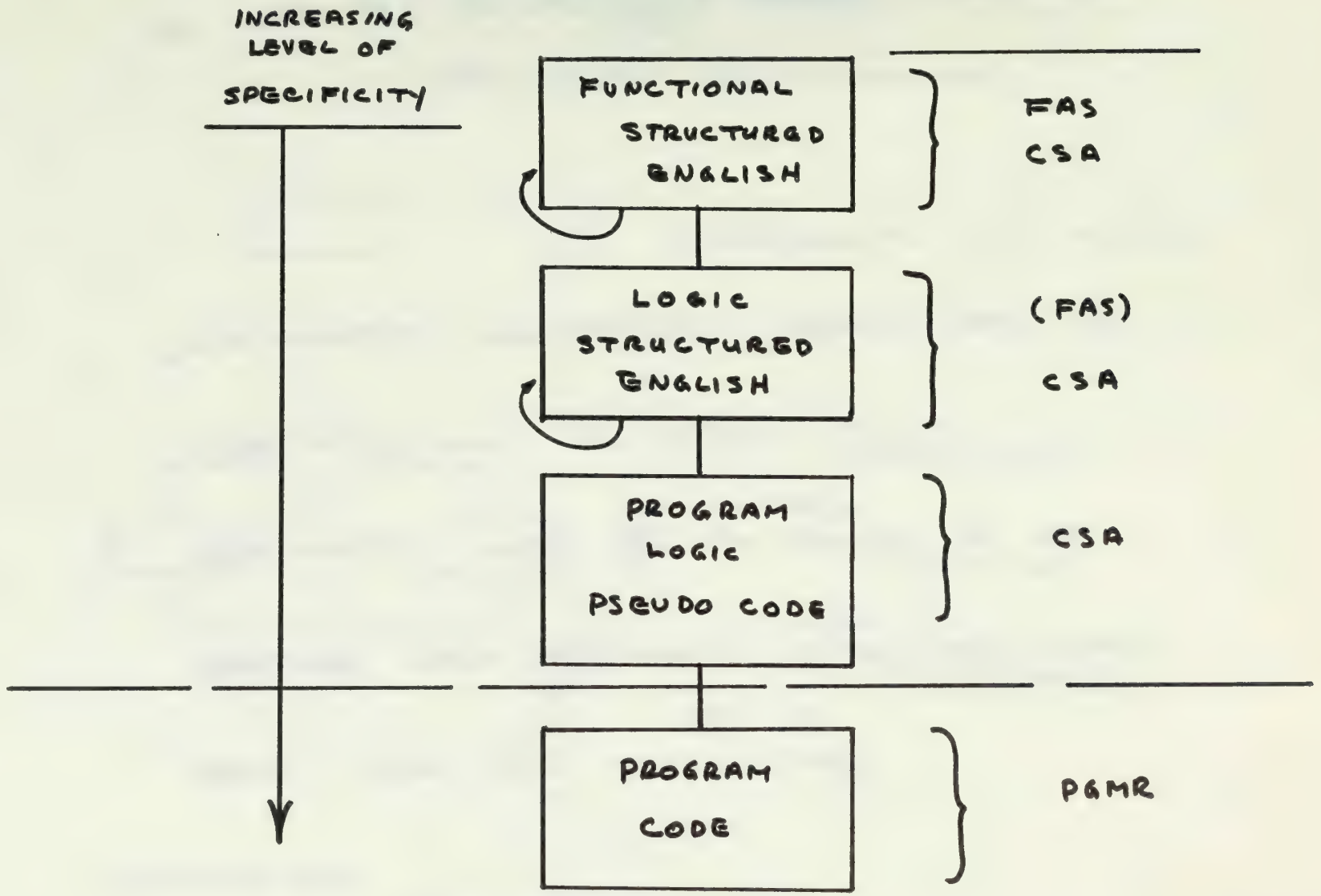
STRUCTURED ENGLISH FORMAT

```
IDENTIFYING NAME
INPUT
    DATA / FILE
OUTPUT
    DATA / FILE
FUNCTIONS
    START (IDENTIFYING NAME)
    (SEQUENTIAL FUNCTIONS, IF ANY)
    IF (SELECTION DECISIONS - TRUE)
        (FUNCTIONS FOR TRUE CASE)
      ELSE (SELECTION DECISIONS- FALSE)
        (FUNCTIONS FOR FALSE CASE)
    END-IF
    (INITIALIZING FUNCTIONS)
    LOOP
      (FUNCTIONS)
    EXIT (EXIT TEST SATISFIED)
```

STRUCTURED ENGLISH FORMATS (CONT.)

```
    (FUNCTIONS)
END-LOOP
    (TERMINATING FUNCTIONS)
END  (IDENTIFYING NAME)
```

STRUCTURED ENGLISH PROCESS



STRUCTURED ENGLISH

VOCABULARY

VERBS: READ, PUT, ACCUMULATE

OBJECTS: DATA DICTIONARY - FILES, DATA FLOWS, --

QUALIFIERS: FROM SET OF NAMES IN DD

* CONJUNCTIONS: IF, UNTIL, WHILE

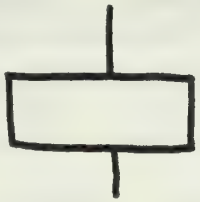
* RELATIONAL ATTRIBUTES: EQUAL, AND, OR

* FROM SET OF RESERVED WORDS

STRUCTURED ENGLISH

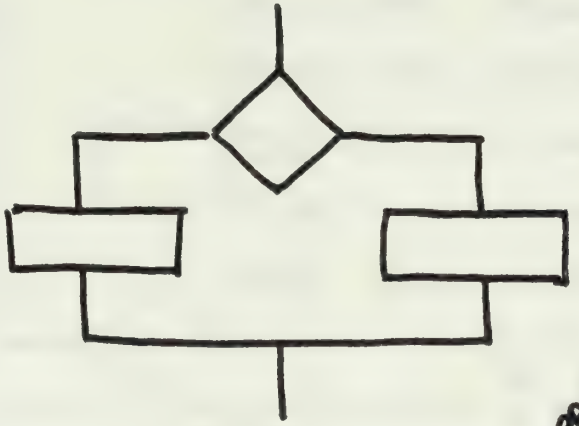
VERSUS

STRUCTURED PROGRAMMING



GOOD

ACCUMULATE YTD TOTAL



USE PARSIMONIOUSLY

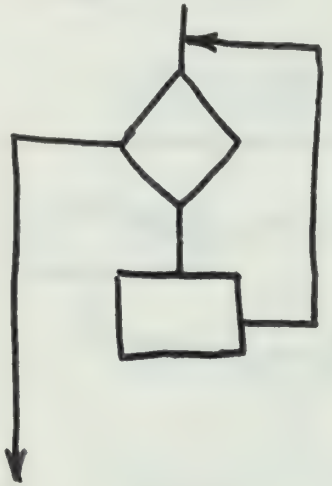
IF (CONDITION)
THEN
(FUNCTION)
ELSE
(FUNCTION)
END-IF

STRUCTURED ENGLISH

VERSUS

STRUCTURED PROGRAMMING

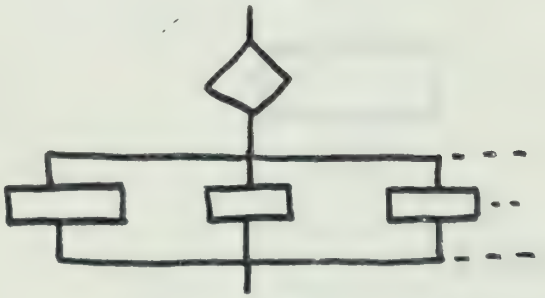
REPETITION



USE WITH CARE

DO WHILE MST-STK-NO
< TRAN-STK-NO

CASE



OK

DEPENDING ON WHICH
CASE APPLIES:

CASE 1 : TRUCK OVER WT. WITH PERMIT

CASE 2 : TRUCK OVER WT. W/O PERMIT

CASE 3 : TRUCK OVER WT WITH I/50

PSEUDOCODE

```
INITIALIZE (OPEN, ZERO COUNTERS, etc)
READ FIRST MASTER
    DO-WHILE THERE ARE MORE MASTERS
        DO-WHILE THERE ARE MORE ACCOUNTS
            ACCUMULATE ACCOUNT-TOTAL
        END DO
        CALCULATE TARE
        CALCULATE CUSTOMER DISCOUNT
        CALCULATE COMMISSION
        PRINT CUSTOMER-LINE
        READ NEXT MASTER
    END-DO
PRINT ACCUMULATED-TOTAL-LINE
TERMINATE PROCEDURES
```

STRUCTURED ENGLISH

ADVANTAGES

1. USED THROUGHOUT THE PROJECT
2. CONCISE, READABLE
3. TAILORED TO SUIT USER
4. TIED TO DATA DICTIONARY
5. CAN BE LEARNED & USED QUICKLY
6. MAY BE AUTOMATED FOR UPDATE

STRUCTURED ENGLISH

DISADVANTAGES

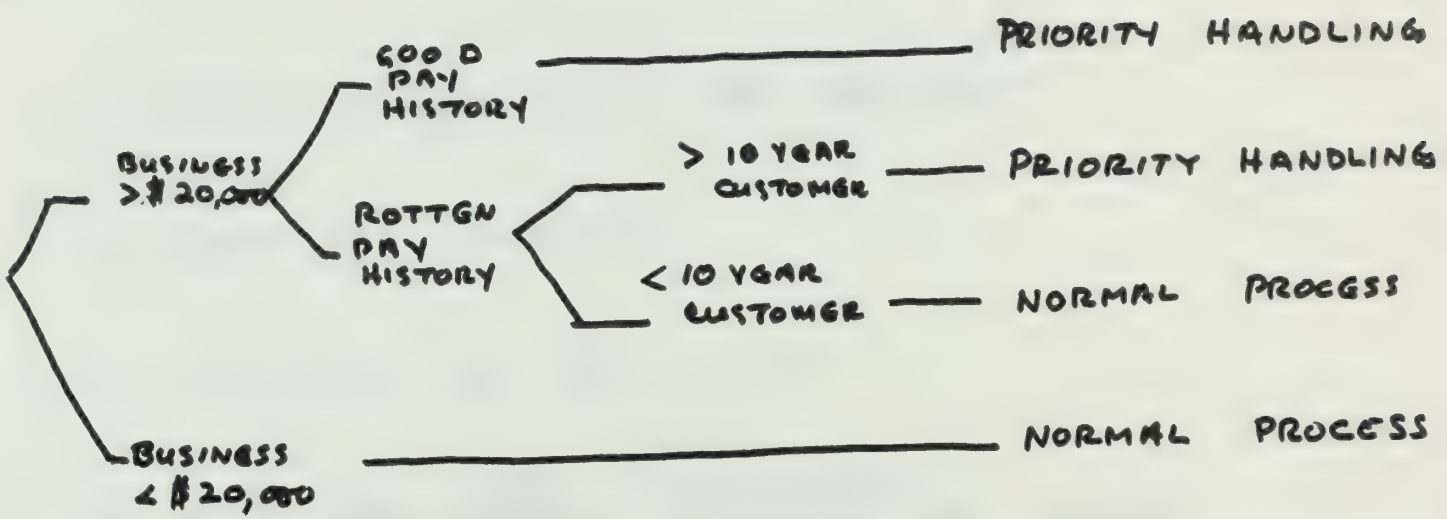
1. TAKES TIME TO DEVELOP

2. MAY SEEM MORE FORMAL THAN IT IS

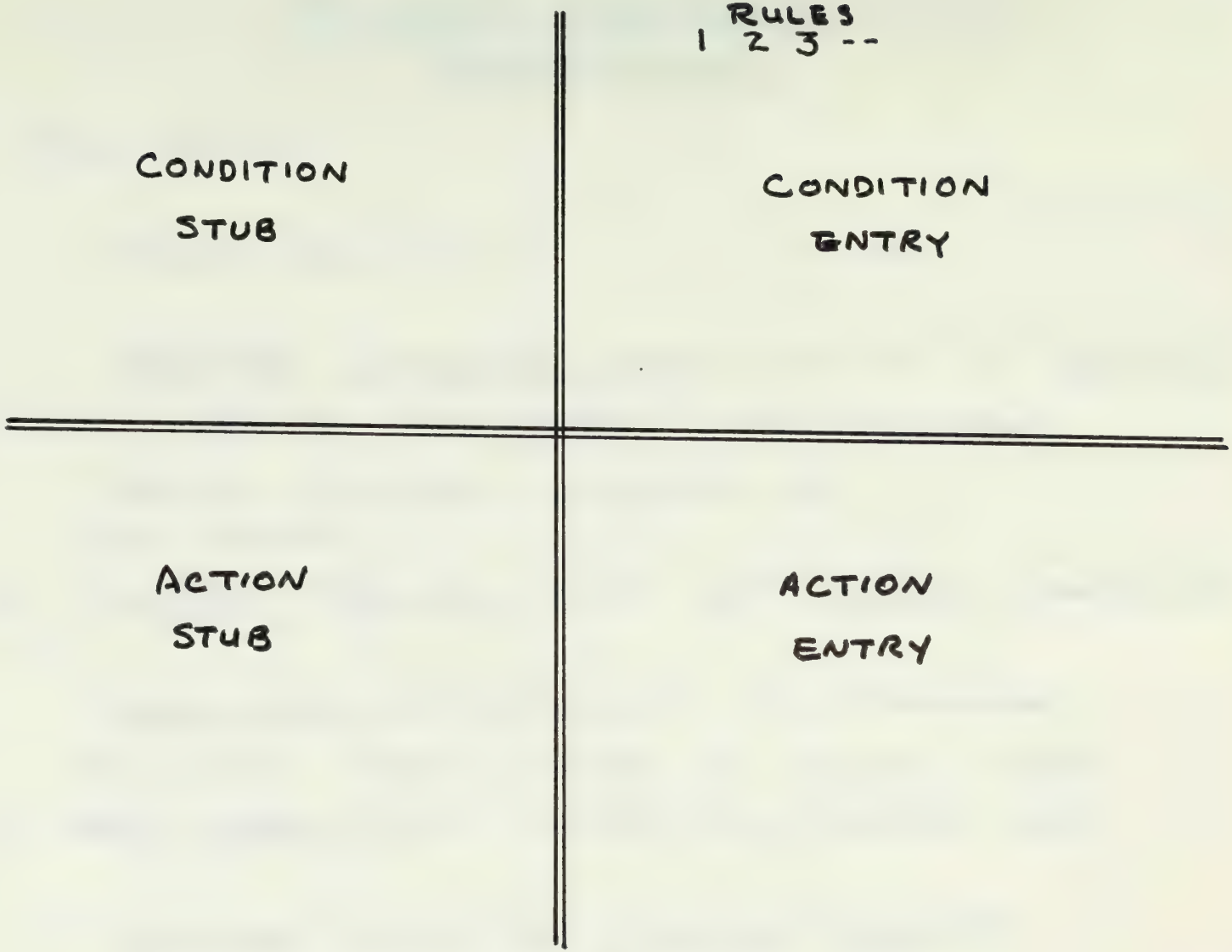
3. MAY FRIGHTEN USER (FUNCTIONAL)

4. MAY GIVE UNDUE IMPRESSION OF TRUTH

DECISION TREE



DECISION TABLE



DECISION TABLE OR DECISION TREE ?

1. USE TREE WHEN NUMBER OF ACTIONS SMALL
   USE TABLE WHEN NUMBER OF ACTIONS LARGE
   AND MANY COMBINATIONS OF CONDITIONS

2. IF TREE DOES NOT SHOW FULL PROBLEM COMPLEXITY,
   USE TABLE

3. IF PROBLEM "TOO LARGE", USE NEITHER.
   CONSIDER FURTHER PARTITIONING OF PROBLEM

PACKAGING FUNCTIONAL DESCRIPTIONS

FINAL DETAILS

ERROR MESSAGES

START UP & SHUT DOWN

SPECIAL CONTROL INFORMATION

SPECIAL FORMATS

CONVERSION

PERFORMANCE

SECURITY

GUIDE TO
FUNCTIONAL DESCRIPTION

COVERS:

ELEMENTS OF A FUNCTIONAL DESCRIPTION

DATA FLOW GRAPH USE & CONVENTIONS

DATA DICTIONARY USE, CONVENTIONS, NOTATIONS

CORRELATION BETWEEN DFG & DATA DEF.

CORRELATION BETWEEN DFG & PROCESS DESCRIP.

INTERPRETATION OF DATA STRUCTURE DIAGRAMS

SUMMARY

STRUCTURED FUNCTIONAL DESCRIPTION

- COMPONENTS—

- DATA DICTIONARY
- DATA FLOW GRAPHS
- FUNCTION DESCRIPTIONS

- CHARACTERISTICS-

- GRAPHIC
- PARTITIONED
- RIGOROUS
- MAINTAINABLE
- ITERATIVE
- LOGICAL

STRUCTURED WALK THROUGHS

DESIGN

TRANSITION

ANALYSIS TO DESIGN

* ANALYSIS - SPECIFYING WHAT ACTIONS AGAINST WHAT DATA AT WHAT TIME

* DESIGN - SPECIFYING HOW REQUIRED ACTIONS ARE TO BE PHYSICALLY, LOGICALLY IMPLEMENTED

DESIGN

DESIGN IS NOT A SUBSTITUTE FOR ANALYSIS

PROBLEM LOGIC & OBJECTIVES MUST BE DOCUMENTED BEFORE BEGINNING DEVELOPING THE SOLUTION

I.E. MUST DEVELOP DETAILED KNOWLEDGE OF WHAT NEEDS TO BE DONE BEFORE ATTEMPTING THE HOW

DESIGN MUST BE KEYED TO FUNCTIONAL DESCRIPTION

DESIGN

GENERAL GOALS

MAINTAINABLE — KEEPING THE SYSTEM RUNNING IN A RELATIVELY CONSTANT ENVIRONMENT

MODIFIABLE - KEEPING THE SYSTEM RUNNING IN A CHANGING ENVIRONMENT

FLEXIBLE - PERFORM VARIOUS FUNCTIONS ON SAME GENERAL THEME

GENERAL - ABILITY OR SCOPE OF SYSTEM

EFFICIENT - QUICK & CHEAP

EFFECTIVE - DO WHAT YOU WANT IT TO

DESIGN

DEFINITION:

THE PROCESS OF USING THE FUNCTIONAL DESCRIPTION TO DEVELOP THE MODEL OF A PHYSICAL SYSTEM TO SATISFY USERS NEEDS

OBJECTIVES

#1 PROVIDE THE USER WITH DESIRED PRODUCTS

ALSO PERFORMANCE

CONTROL

MAINTAINABILITY

DESIGN

PERFORMANCE

EXPRESSED IN TERMS OF:

THROUGHPUT - TRANSACTIONS PER HOUR

RUN TIME - TIME TO PROCESS FIXED AMOUNT OF DATA IN BATCH MODE

RESPONSE TIME - TIME BETWEEN PUSHING START & RECEIPT OF OUTPUT

MENU OF ALTERNATIVES

1. BATCH

   INEXPENSIVE

   SLOW WORKING (12-24- HOURS)

   ERROR PROBLEMS (TURNAROUND)

2. SOURCE DATA ENTRY - ON LINE BATCH

   ON-LINE ERROR CORRECTION BY PEOPLE WHO KNOW; WITH IMPROVED TIMELINESS

3. ON LINE DATA ENTRY

   IMMEDIATE UPDATE

   ON LINE ENQUIRY

   RELATIVELY EXPENSIVE

   DATA STORE UPDATED TO LAST ENTERED TRANSACTION

MENU OF ALTERNATIVES
(CONT.)

4. DISTRIBUTED

   LOCAL FILES LOCALLY MANAGED

   FAIRLY INEXPENSIVE (MINI/MICRO COMPUTERS)

   "SYSTEM" LESS LIKELY TO STOP ENTIRELY

5. DEDICATED MINICOMPUTERS

   FOR UNIQUE APPLICATIONS WITH HIGH SERVICE LEVELS

   OVERALL MANAGEMENT PROBLEMS

6. IMPROVED MANUAL SYSTEM

   CHANGE OF PRESENT { PERSONNEL / ORGANIZATION / EQUIPMENT } TO REMOVE NECESSITY OF AUTOMATION.

DESIGN

PERIPHERALS AFFECTING PERFORMANCE:

1. MANDATORY, UNREALISTIC DATES OF COMPLETION
2. MEMORY RESTRICTIONS
3. RUN TIME OF PROGRAM

DESIGN

FACTORS AFFECTING THROUGHPUT

1. NUMBER OF INTERMEDIATE FILES IN SYSTEM

2. NUMBER OF TIMES A FILE IS PASSED

3. NUMBER OF DISK FILE SEEKS

4. TIME SPENT ON SYSTEM OVERHEAD, CALLING PROGRAM/MODULE/PAGES.

5. ACTUAL TIME TO EXECUTE A PROGRAM

DESIGN

CONTROL CONSIDERATIONS

1. CHECK DIGITS

2. BATCH TOTALS

3. JOURNALS & AUDIT TRAILS

4. LIMITING FILE ACCESS

5. CHECK PROGRAMS

   INTER-FILE BALANCING

DESIGN

CHANGEABILITY

REQUIRED

MODULARITY HELPS → COUPLING - LOOSE

COHESIVENESS - LOTS

BLACK BOX CONCEPT

DESIGN

- PHASED
- INCREMENTAL
- BOTTOM - UP
- TOP DOWN

DESIGN

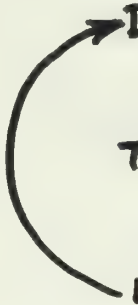
PHASED APPROACH

- DESIGN, CODE, TEST EACH MODULE BY ITSELF (UNIT TEST)
- COMBINE ALL MODULES FOR A "SYSTEMS INTEGRATION TEST"
- FIELD TEST

DESIGN

INCREMENTAL APPROACH

DESIGN, CODE, TEST A MODULE



DESIGN, CODE, TEST A SECOND MODULE

TEST & DEBUG THE COMBINATION

LOOP

DESIGN

BOTTOM UP

UNIT TESTING (MODULE → PROGRAM)

SUB-SYSTEM TESTING (PROGRAM → APPLICATION)

SYSTEM TESTING

DESIGN

TOP DOWN

A STRATEGY

DEFINES MAIN OR ROOT MODULES FIRST

ESSENTIALLY MEANS PROCEEDING BY REFINEMENT
FUNCTIONS AT EACH LEVEL DEFINED IN TERMS OF SUBORDINATE FUNCTIONS

STRUCTURED DESIGN INCLUDES:

CHARTING TECHNIQUES

COUPLING

COHESION

HEURISTICS

TRANSFORM-CENTERED DESIGN

STRUCTURED DESIGN INCLUDES:

TRANSACTION ANALYSIS

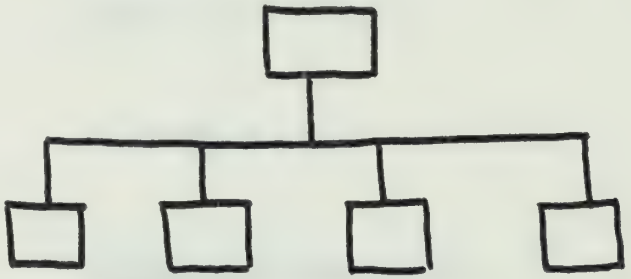
COMMUNICATION

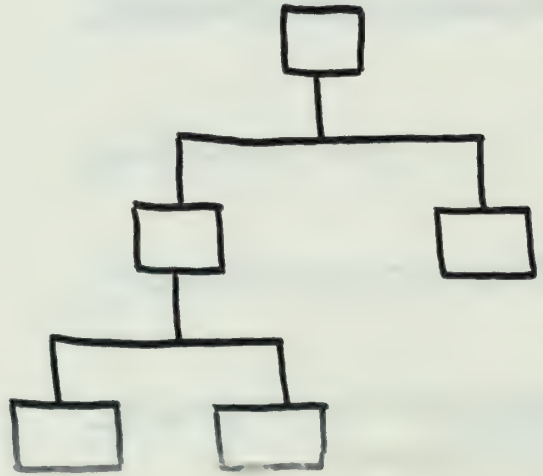
PACKAGING

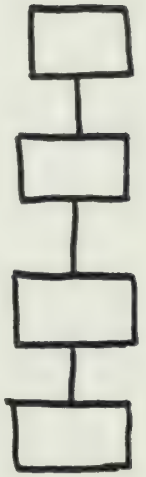
IMPLEMENTATION

OPTIMIZATION

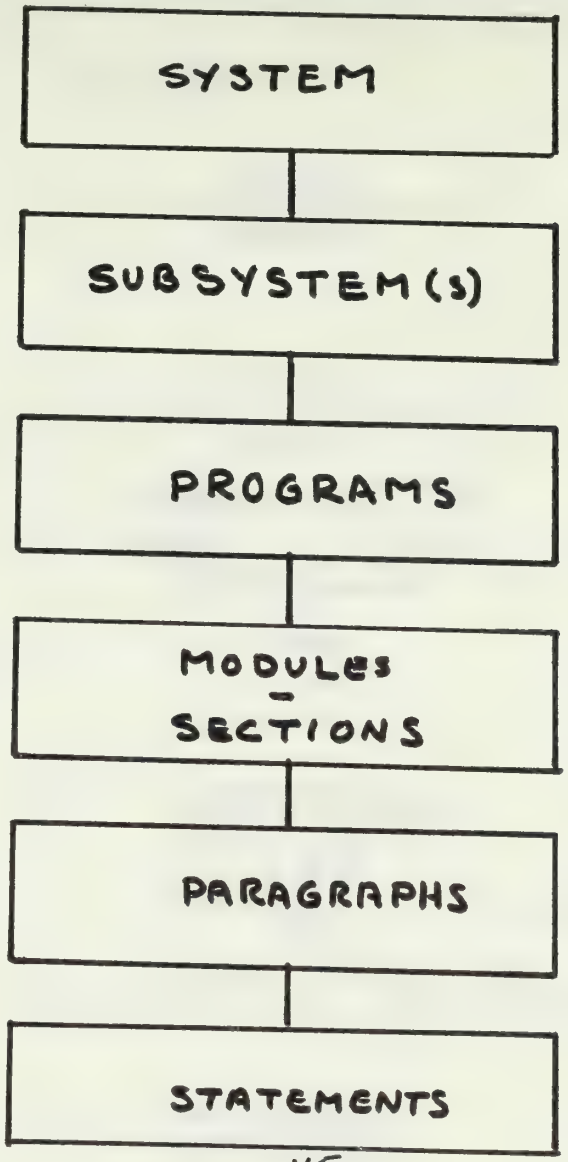
MANAGEMENT PARALLELS WITH SYSTEM DESIGN





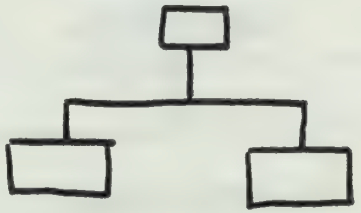


THE HIERARCHICAL STRUCTURE



MODULES

BUILDING BLOCKS OF THE SYSTEM



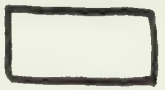
BLACK BOX

ENGINEERING CONCEPT

GOOD

FAIR

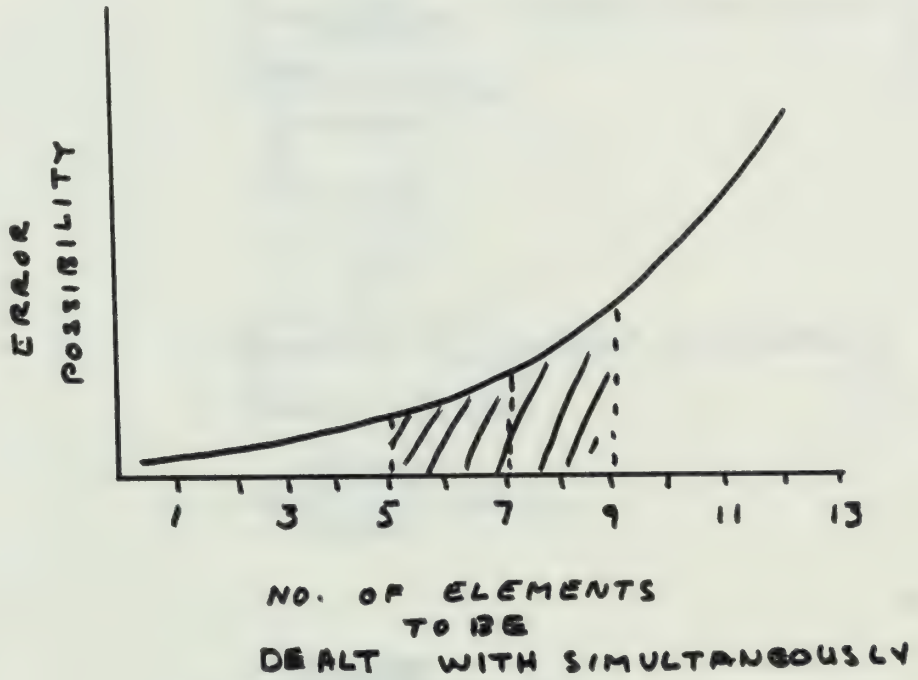
{ ONLY IF N
WHOLE N, EVEN, < 4 DIGITS }



BAD

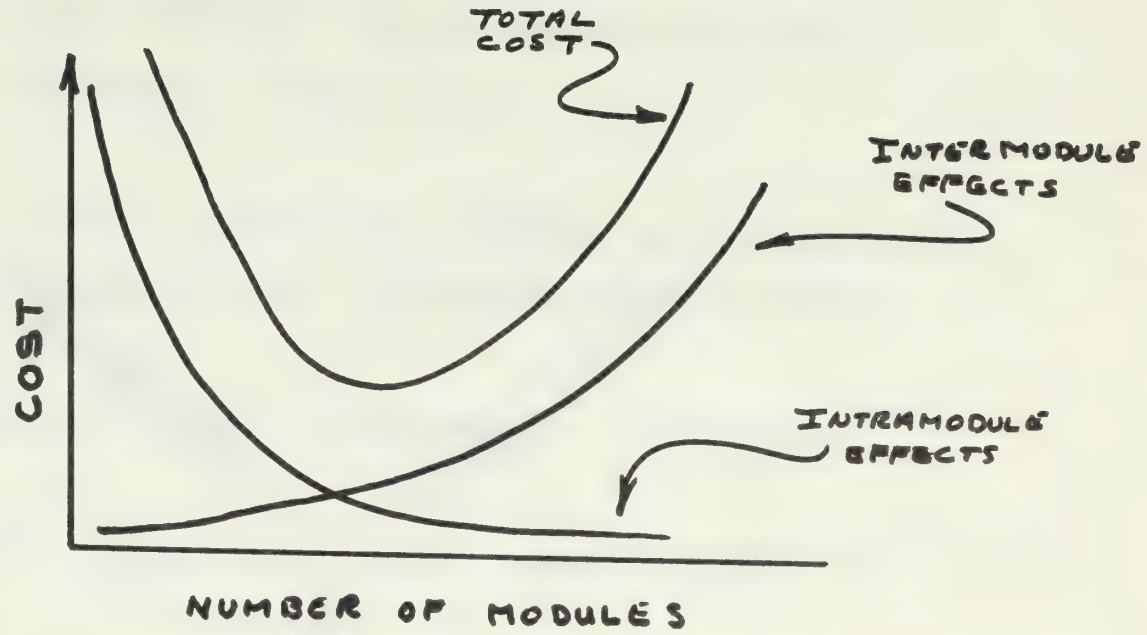
{ ONLY IF N
WHOLE N, EVEN, < 4 DIGITS,
NOT PRIME N }

THG MAGIC NUMBER 7 (±2)*



*PSYCHOLOGY REVIEW - 1956

INTER/INTRA MODULE EFFECTS



MODULE COMPONENTS

MODULE

PURPOSE: RECEIVE INPUT, PERFORM ONE OR MORE TRANSFORMATIONS, RETURN RESULTS

CONSISTS OF ONE OR MORE STATEMENTS WITH THE FOLLOWING CHARACTERISTICS:

- STATEMENTS ARE LEXICALLY TOGETHER
- HAS AN IDENTIFIABLE BOUNDARY
- CALLED OR REFERENCED BY A NAME
- CAN BE CALLED FROM ANY PART OF THE PROGRAM

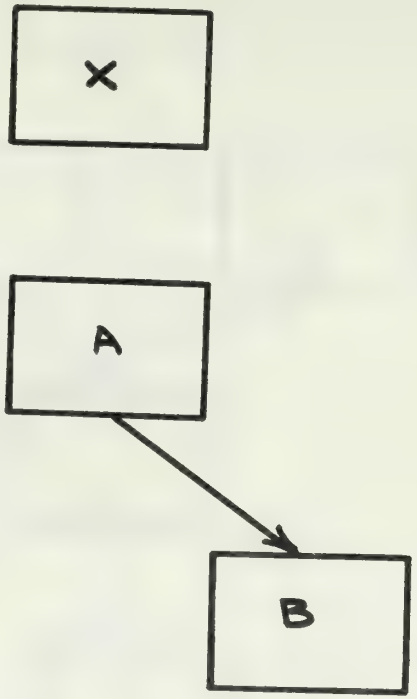
PROGRAM DESIGN SUMMARY

- BUILD A FUNCTIONAL, MODULAR STRUCTURE
- IMPLEMENT
  - IN MODULES USING CALLS, ETC

    OR

  - IN ONE PROGRAM USING PERFORM, DO, ..
- RESULTS SHOULD BE
  - SMALL MODULES
  - MORE RELIABLE SYSTEMS
  - EASIER MAINTENANCE
  - EASIER EXPANSION
  - EASIER TESTING
  - REUSABLE CODE

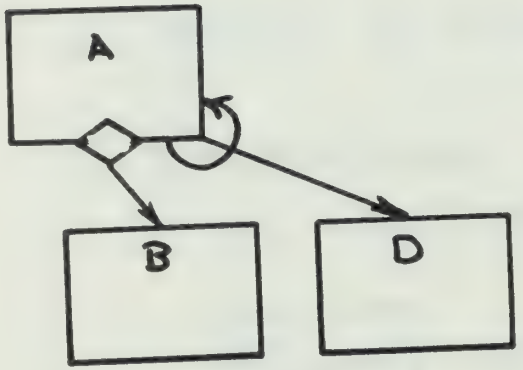
STRUCTURE CHART NOTATIONS



ANY MODULE

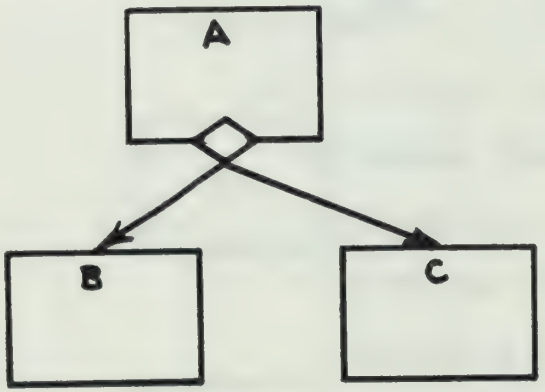
MODULE A CALLS/INVOKES MODULE B

B IS SUBORDINATE TO A



*CONDITIONAL CALL IN A MODULE TO B MODULE

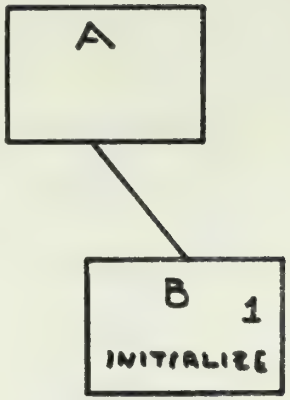
*MODULE A CALLS D MULTIPLE TIMES



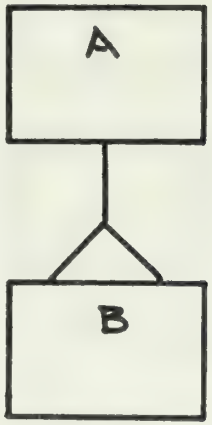
*CONDITIONAL CALL TO MODULE B OR C

{ IF THEN ELSE
CASE }

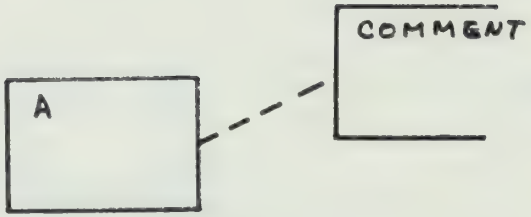
* USE SPARINGLY

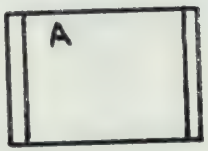


MODULE B CALLED ONE TIME ONLY



LEXICAL INCLUSION. B IS ACTUALLY WRITTEN WITHIN THE LEXICAL BOUNDARIES OF A

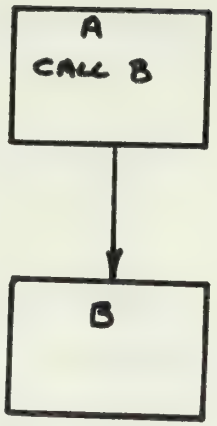




PREDEFINED MODULE

OPERATING ENVIRONMENT WITHIN WHICH THE PROGRAM RUNS. MAY INCLUDE MONITOR, HARDWARE, ETC

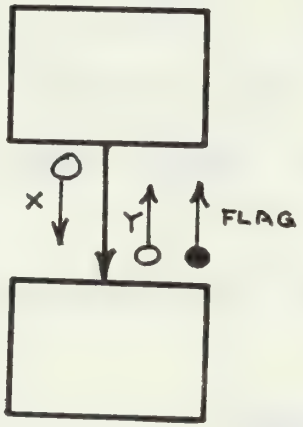
CONNECTION GRAPHICS



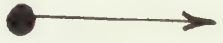
AN INTERMODULE REFERENCE
TO A MODULE IDENTIFIER

FUNCTIONS OF AN INTERFACE:

- TRANSMIT DATA INTO MODULE
- TRANSMIT DATA OUT OF MODULE
- RECEIVE CONTROL
- TRANSMIT CONTROL



DATA REFERENCE

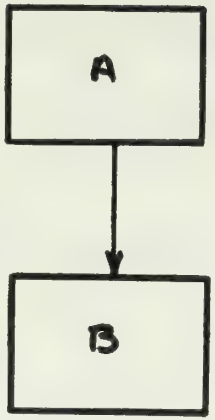


CONTROL REFERENCE

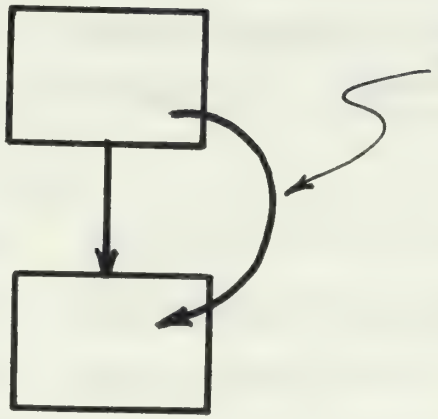
STRUCTURE CHART GUIDELINES

- EACH BOX IS A SINGLE FUNCTION MODULE
- ALL DATA NEEDED BY A MODULE ARE SHOWN EXPLICITLY
- EACH MODULE HAS NECESSARY DATA & SUBORDINATES REQUIRED TO IMPLEMENT THE REQUIRED FUNCTION
- VERIFY CORRESPONDENCE TO DFG FOR POSSIBLE MISSING PROCESSES

MODULE CONNECTIONS



NORMAL - MODULE A REFERENCES B BY NAME



PATHOLOGICAL - REFERENCE TO AN IDENTIFIER WITHIN ANOTHER MODULE

STRUCTURE CHARTS

SHOW

- PARTITIONING
- HIERARCHY & ORGANIZATION
- COMMUNICATION BETWEEN MODULES

DON'T USUALLY SHOW OR EMPHASIZE

- EXECUTION SEQUENCE
- DECISIONS (CASE)
- ITERATIONS

COUPLING

- IS A MEASURE OF THE AMOUNT OF MODULE INTERDEPENDENCE

- CLOSELY RELATED TO COHESION

- INFLUENCED BY:

  TYPE OF CONNECTION BETWEEN MODULES
  COMPLEXITY OF THE INTERFACE
  TYPE OF INFORMATION BEING COMMUNICATED

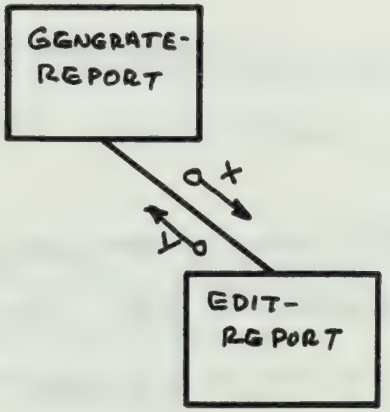
CONCEPT

- MINIMIZING CONNECTIONS

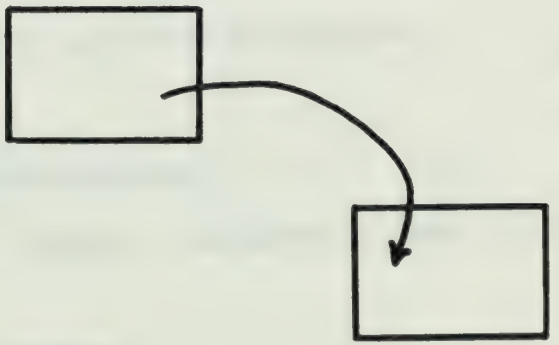
- MINIMIZING "RIPPLE EFFECTS"

MODULE COUPLING

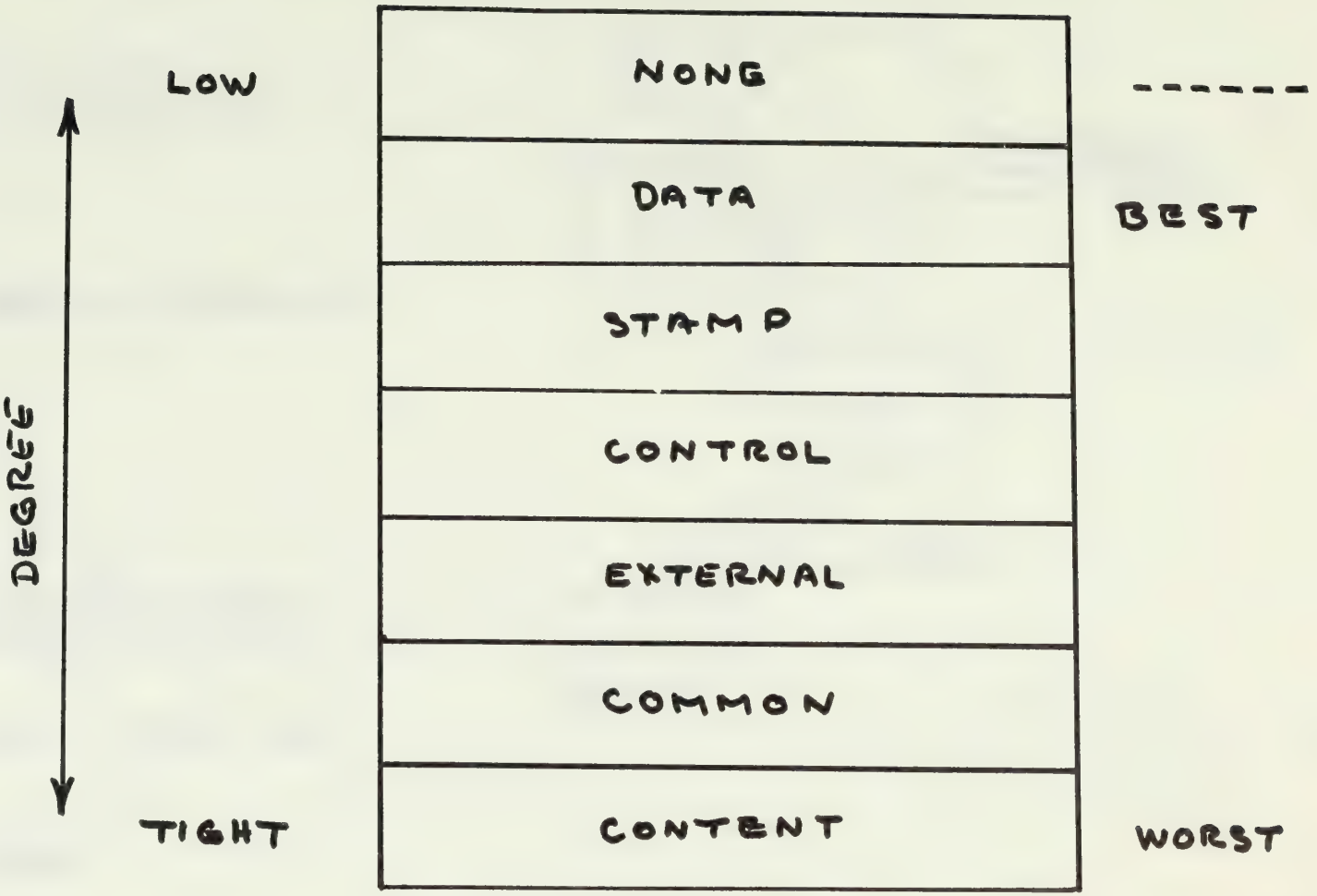
NORMAL



PATHOLOGICAL



COUPLING - TYPES

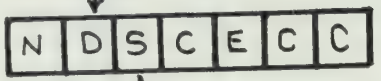


5-257

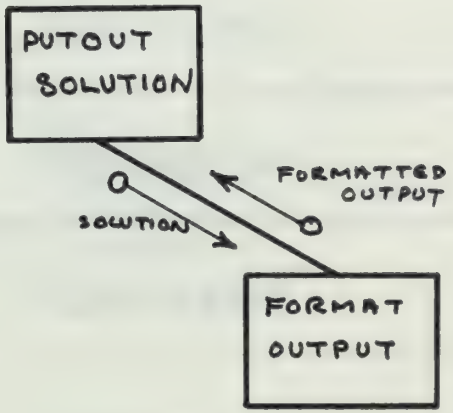
MODULE COUPLING

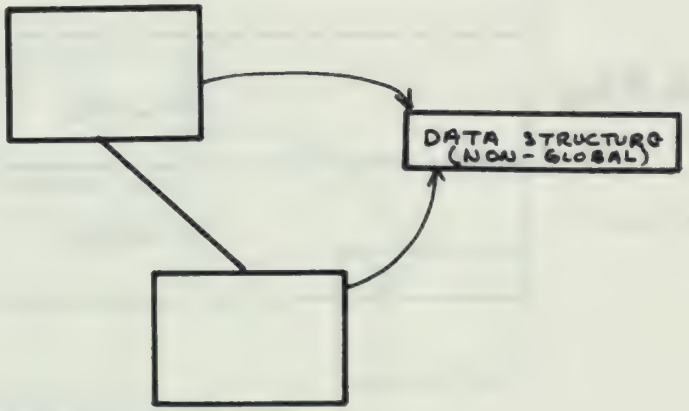
NONE

DATA COUPLING



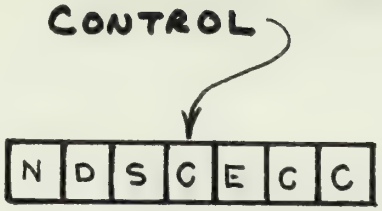
STAMP COUPLING

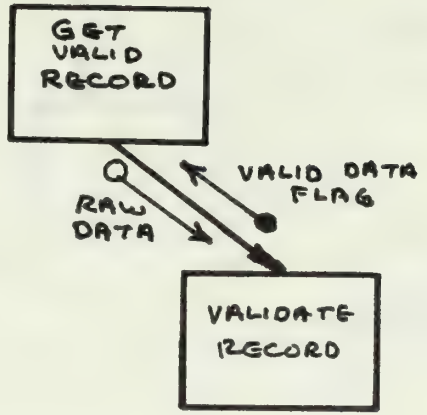




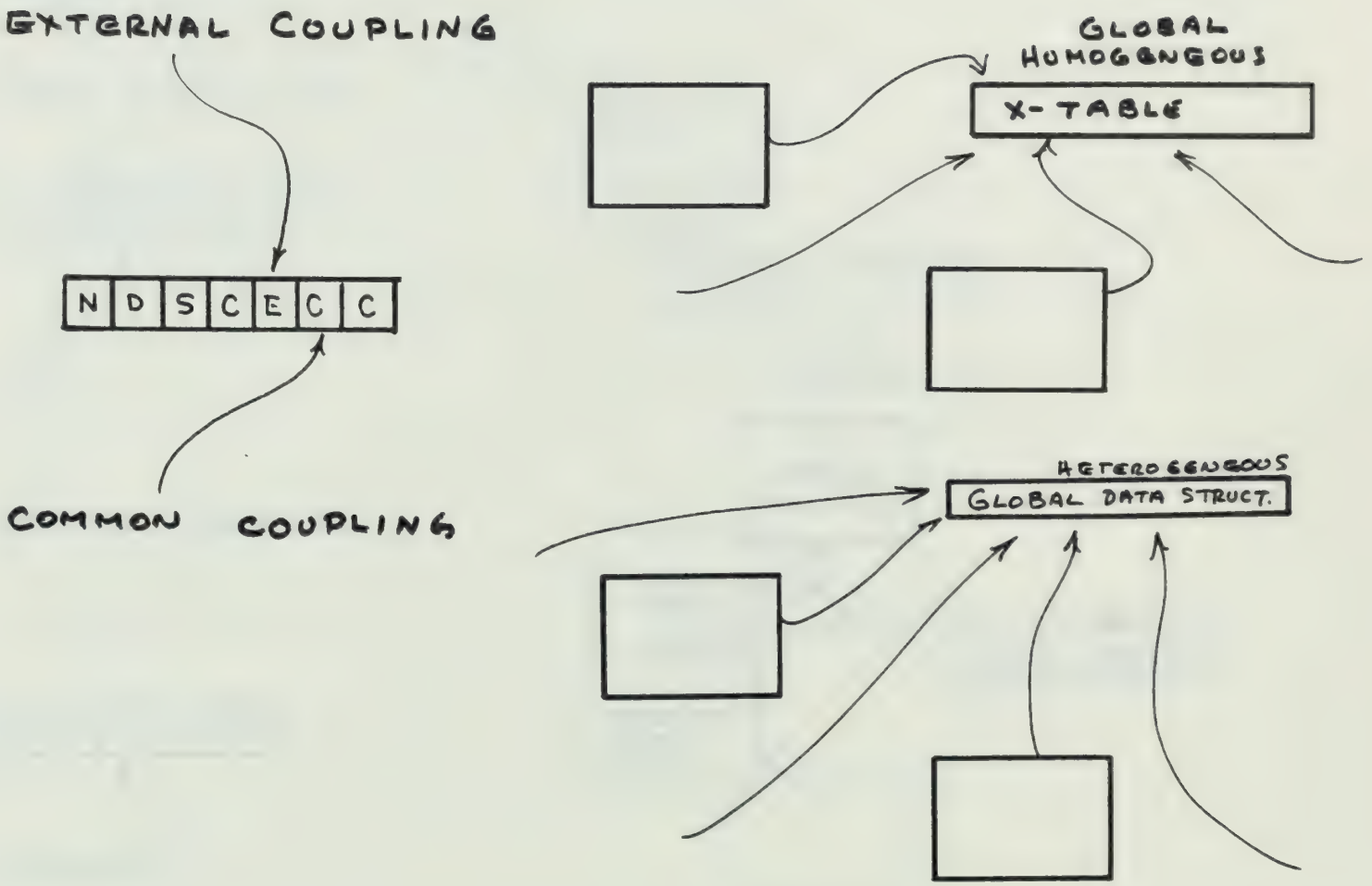
5-25F

MODULE COUPLING

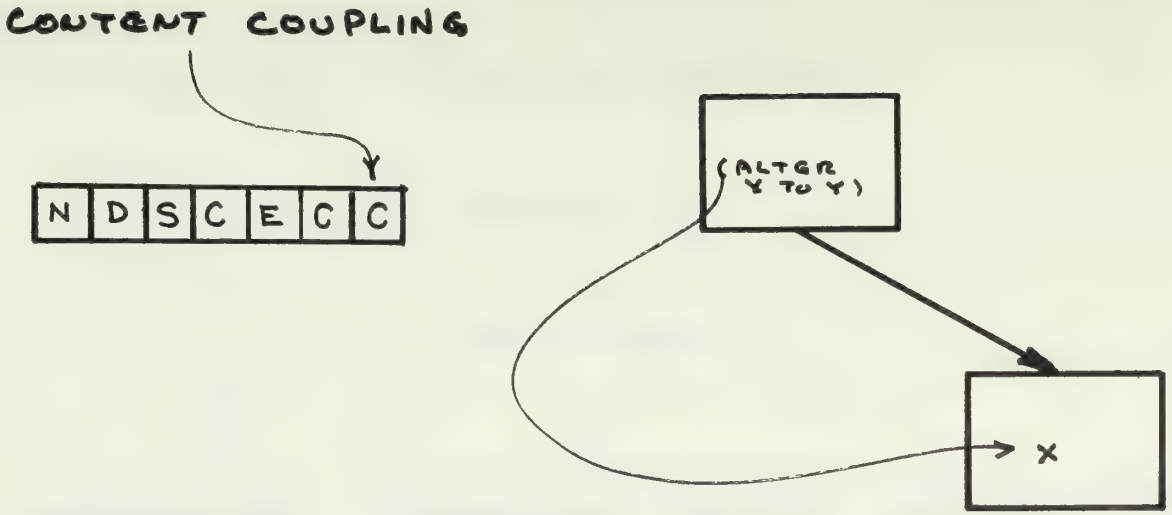




MODULE COUPLING



MODULE COUPLING



COHESION

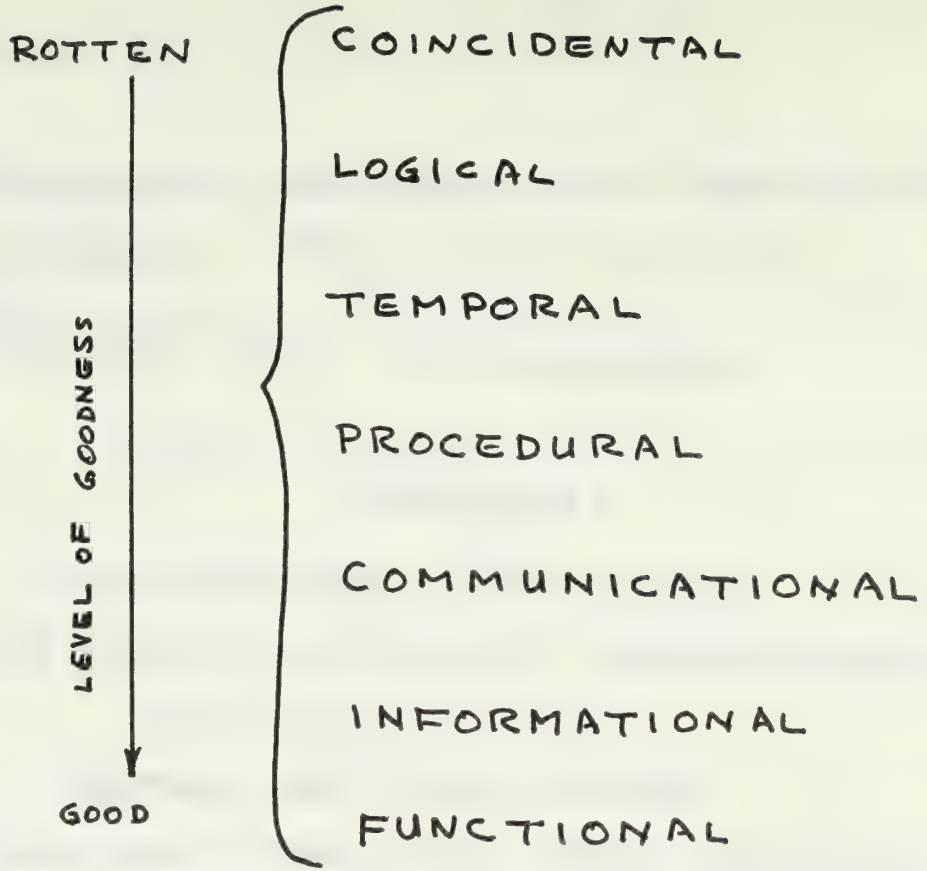
CONSIDERS EACH MODULE IN ISOLATION

NON-QUANTITATIVE MEASURE OF THE INTERNAL BONDING OF ELEMENTS

TIES IN CLOSELY WITH COUPLING

COHESIVE MODULES TEND TOWARD LOW COUPLING

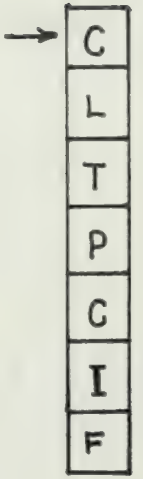
COHESION


ROTTEN
LEVEL OF GOODNESS
GOOD
COINCIDENTAL
LOGICAL
TEMPORAL
PROCEDURAL
COMMUNICATIONAL
INFORMATIONAL
FUNCTIONAL

COHESION

COINCIDENTAL

FUNCTION CAN NOT BE DESCRIBED
EXCEPT BY THE LOGIC



PERFORMS MULTIPLE, COMPLETELY
UNRELATED FUNCTIONS

PROBLEMS

MODULES NOT INDEPENDENT

PROBLEM WITH RE-USABILITY

DEGRADE PROGRAM MAINTAINABILITY

COHESION

LOGICAL

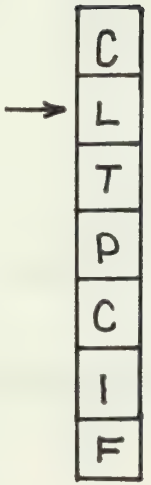
MODULES WITH RELATED SIMILAR FUNCTIONS

EXAMPLES:

GET - ALL

COMPUTE

PUT - OUT



COHESION

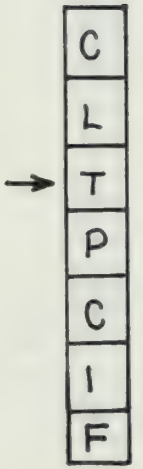
TEMPORAL

ELEMENTS RELATED BY TIME

START-UP

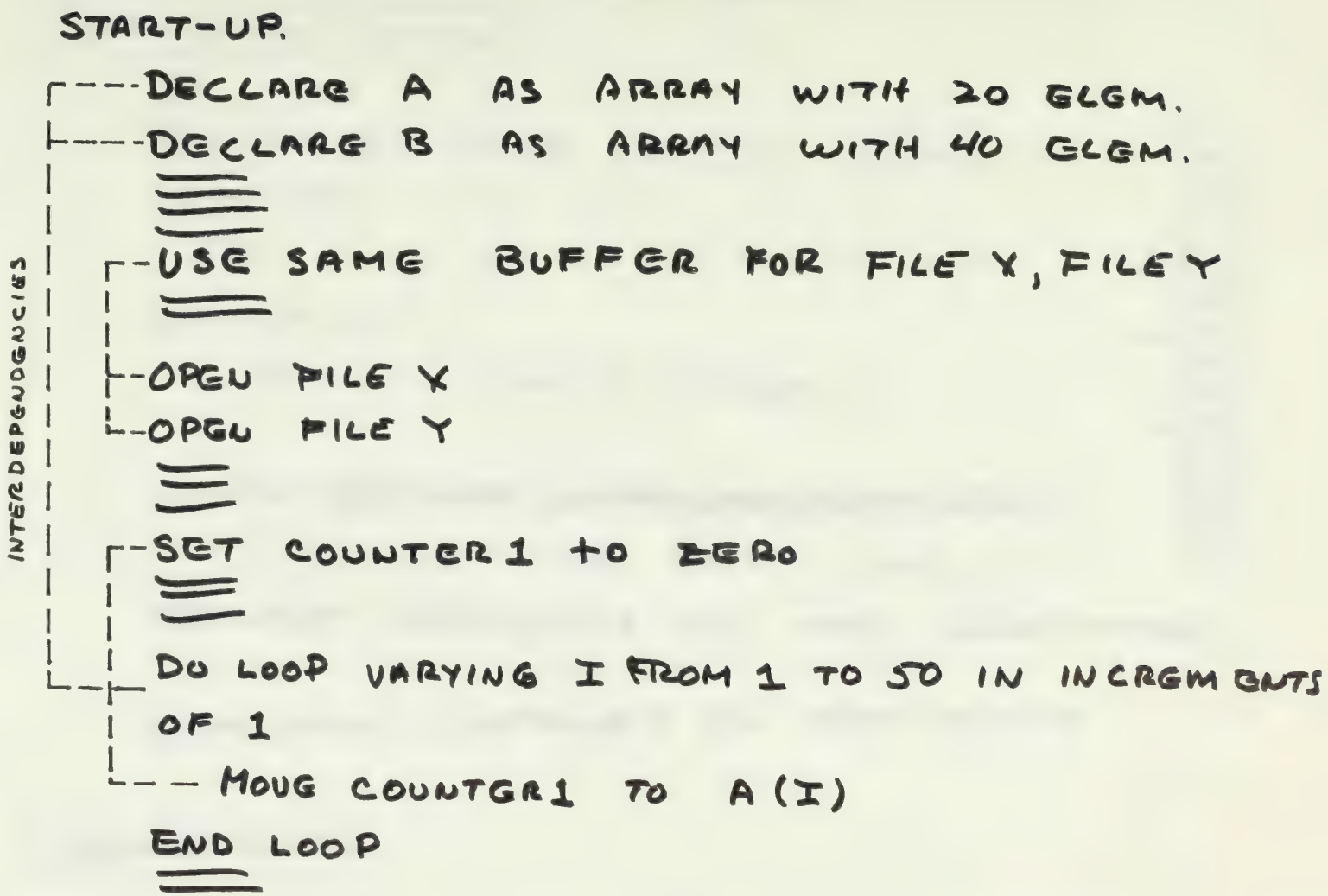
TERMINATION

PROBLEM - INTERTWINING OF CODE



COHESION PROBLEM

(TEMPORAL)


START-UP.
DECLARE A AS ARRAY WITH 20 ELEM.
DECLARE B AS ARRAY WITH 40 ELEM.
USE SAME BUFFER FOR FILE X, FILE Y
OPEN FILE X
OPEN FILE Y
SET COUNTER 1 TO ZERO
DO LOOP VARYING I FROM 1 TO 50 IN INCREMENTS OF 1
MOVE COUNTER1 TO A(I)
END LOOP
INTERDEPENDENCIES

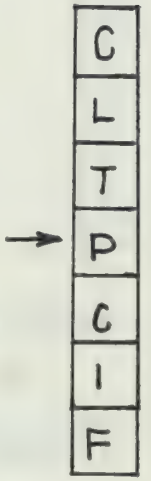
COHESION

PROCEDURAL

FUNCTIONS RELATED BY SEQUENCE

INDIVIDUAL FUNCTIONS NOT GOOD "STAND ALONES"

OFTEN DERIVED FROM FLOW CHARTS



COHESION

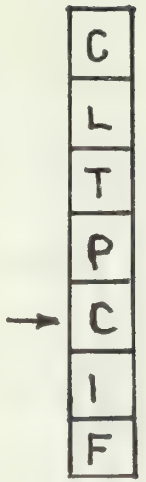
COMMUNICATIONAL

FUNCTIONS SHARE COMMON INPUT OR OUTPUT

EXAMPLES:

PRINT & PUNCH RECORDS

ACCEPT DATA FROM SEVERAL INPUTS & ASSEMBLE INTO PRINT LINE



COHESION

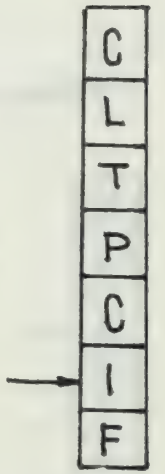
INFORMATIONAL

CONTAINS MULTIPLE ENTRY POINTS

EACH PERFORMS SINGLE, SPECIFIC FUNCTION

ALL FUNCTIONS RELATED BY CONCEPT, DATA STRUCTURE OR RESOURCE HIDDEN WITHIN MODULE

NO CONTROL / DATA FLOW BETWEEN MODULE FUNCTIONS



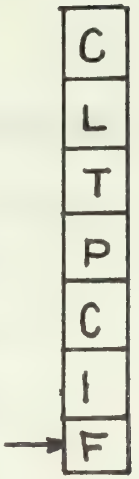
COHESION

FUNCTIONAL

EVERY ELEMENT IS AN INTEGRAL, NECESSARY PART OF THE TARGET FUNCTION

OR

A FUNCTIONAL MODULE IS NOT COMMUNICATIONAL, PROCEDURAL, TEMPORAL, LOGICAL, OR COINCIDENTAL.



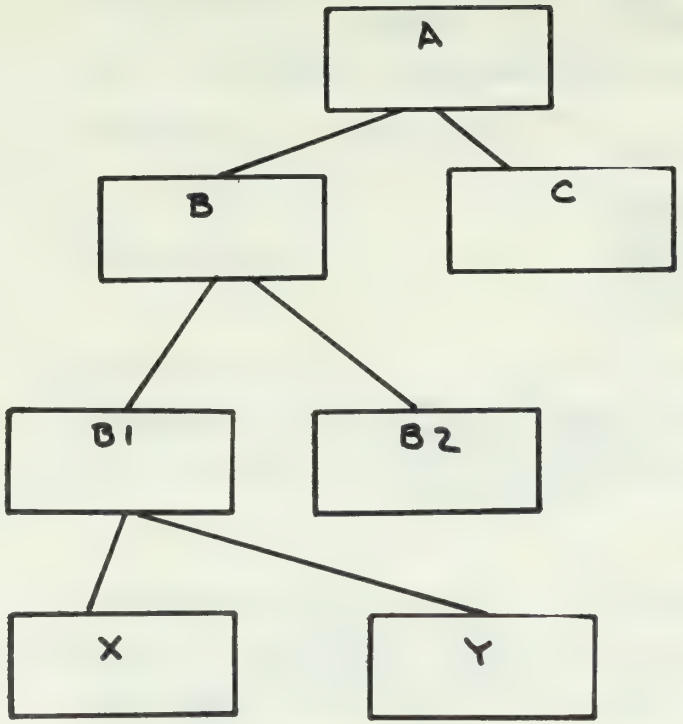
RELATIONSHIP: COUPLING & COHESION

COHESION IS THE MOST IMPORTANT CONCEPT. AIM FOR HIGH COHESION & LOW COUPLING WILL GENERALLY RESULT

LOW LEVELS OF COHESION CAN BE EXPECTED TO INCREASE COUPLING AS MORE FLAGS & SWITCHES ARE INTRODUCED IN THE CODE

SINCE EACH PORTION OF A LOW COHESION MODULE IS RELATED TO SOME OTHER PART OF THE SYSTEM, THIS LEADS TO HIGHER COUPLING

MORPHOLOGY FEATURES : WIDTH & DEPTH



DEPTH OF 4.

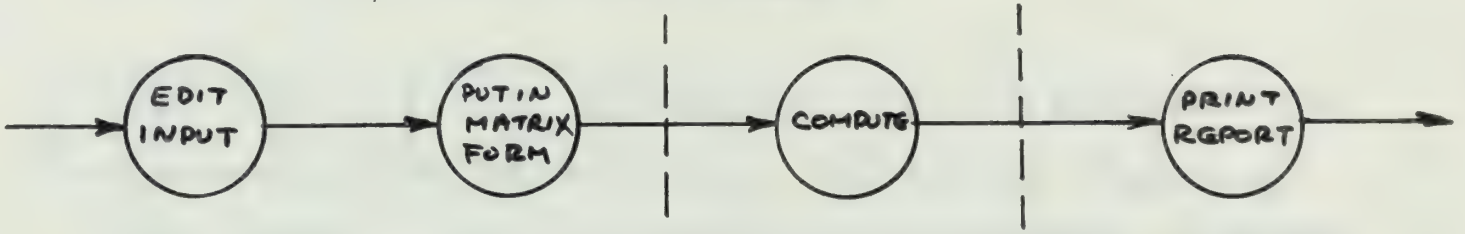
GOOD OR BAD?

DEPTH ITSELF NOT A VALID MEASURE OF GOODNESS OF DESIGN

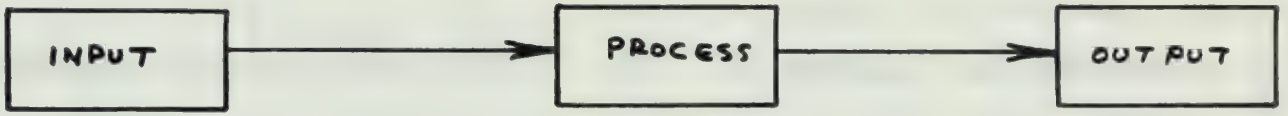
MORPHOLOGY

SIMPLE SYSTEMS

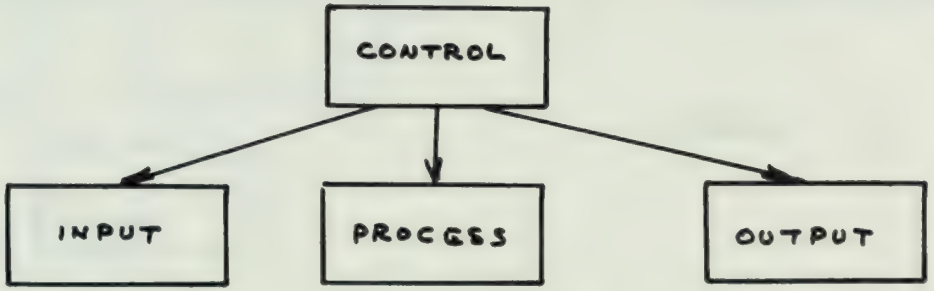
MORPHOLOGY : SHAPE OF SYSTEM WITH RESPECT TO STRUCTURE.



SIMPLE PROCESS MODEL



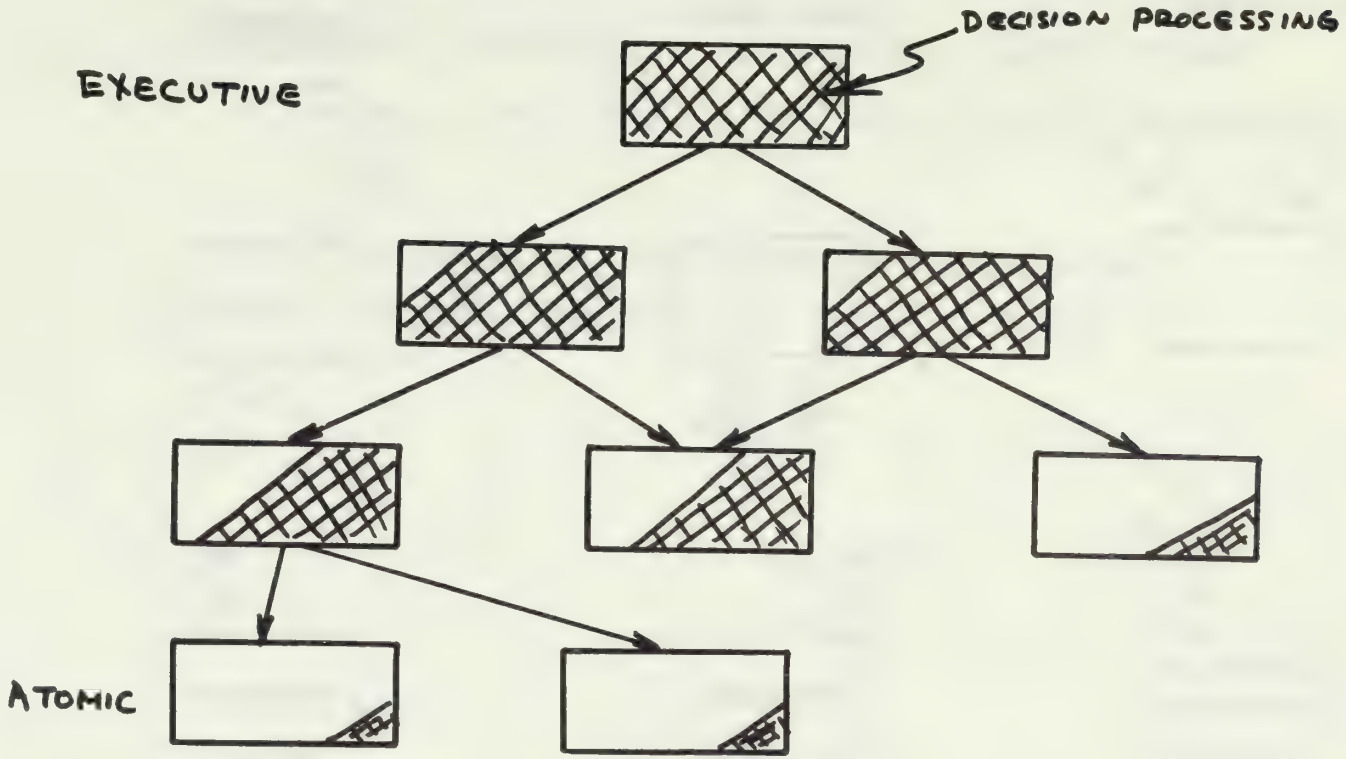
IPO VERSION



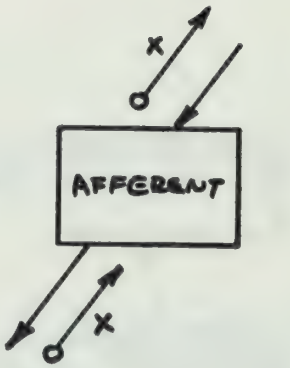
CIPO VERSION

FACTORING

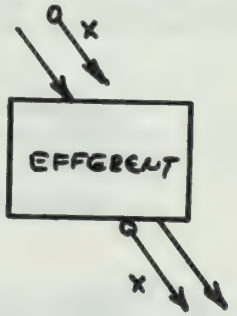
A SYSTEM IS SAID TO BE COMPLETELY FACTORED IF ALL ACTUAL PROCESSING IS DONE BY BOTTOM LEVEL "ATOMIC" MODULES



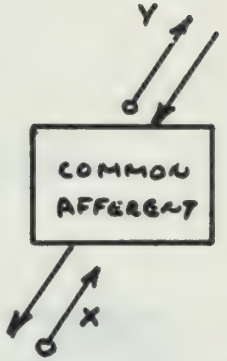
TYPES OF FLOWS



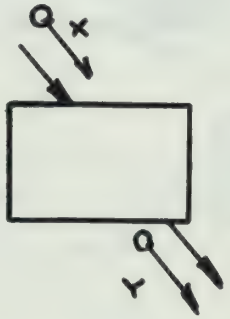
AFFERENT



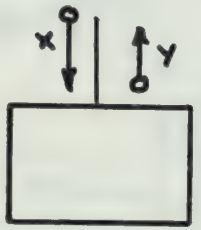
EFFERENT



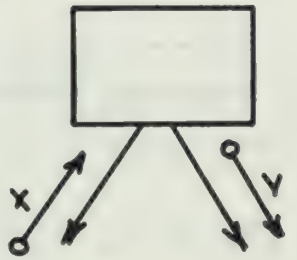
COMMON AFFERENT



COMMON EFFERENT



TRANSFORM



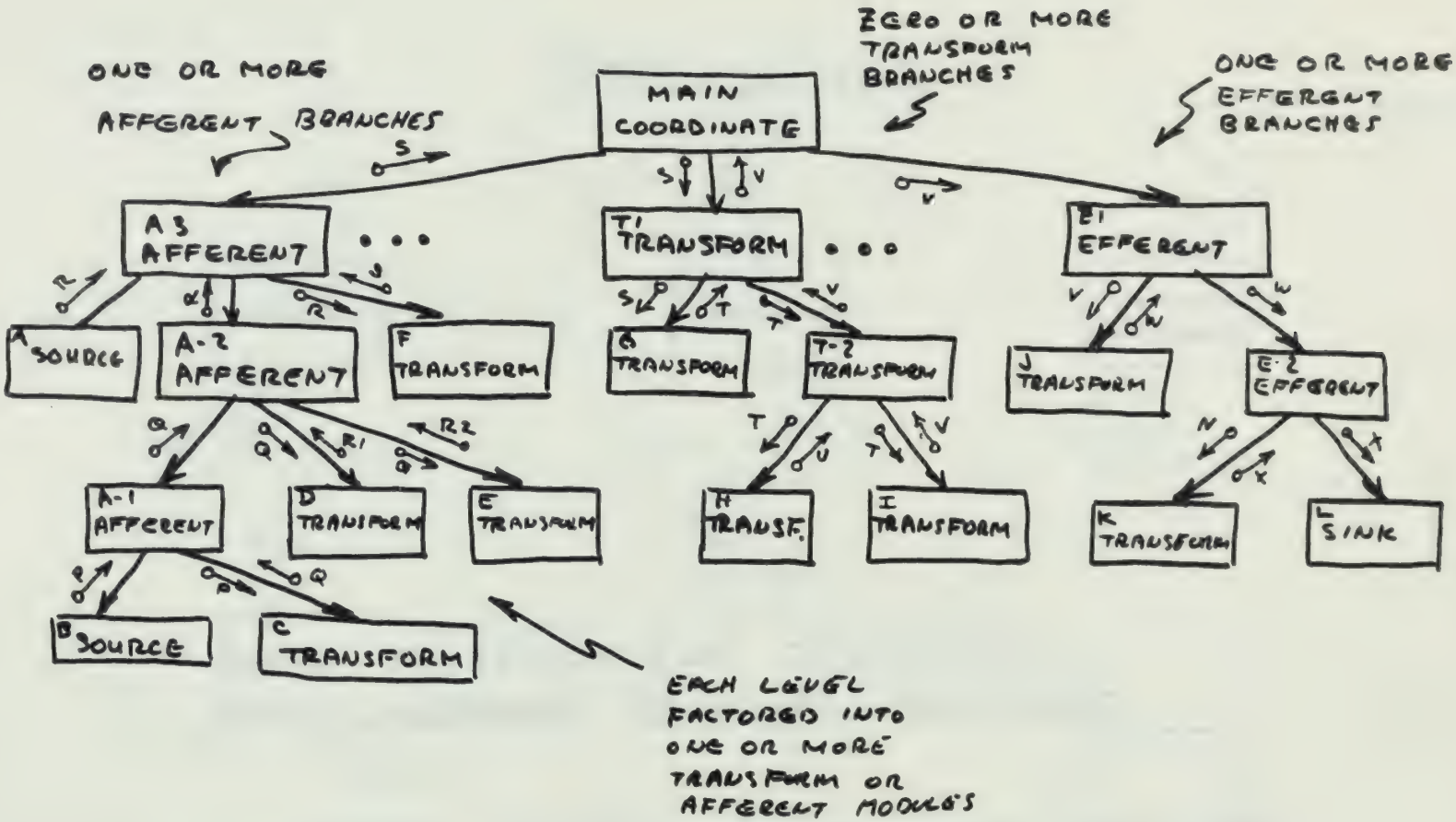
COORDINATE

TRANSFORM CENTERED

HIGHLY FACTORED

AFFERENT, EFFERENT BRANCHES BALANCED
(MODEL NEITHER INPUT OR OUTPUT DRIVEN)

BOTH AFFERENT, EFFERENT BRANCHES
HAVE CHARACTERISTIC STRUCTURE



TRANSFORM CENTERED MODEL

MORPHOLOGY

FAN OUT

FAN IN

"MOSQUE" SHAPE

INPUT DRIVEN

OUTPUT DRIVEN

TRANSFORM CENTERED

DESIGN HEURISTICS

I MODULES SHOULD BE OF REASONABLE SIZE

REASONABLE IS ?

10 - 20 STATEMENTS

ONE PAGE LISTING

ONE WEEKS WORK

HEURISTIC

II SYSTEM SHOULD HAVE REASONABLE SPAN OF CONTROL : 7 ± 2

DEFINITION:

SPAN OF CONTROL - REFERS TO NUMBER OF IMMEDIATE SUBORDINATE MODULES

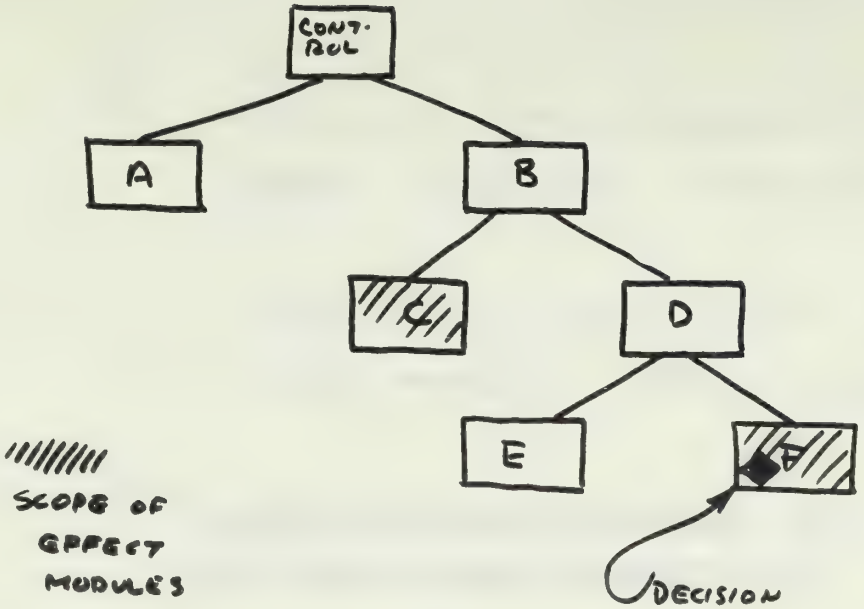
HEURISTIC

III AVOID SCOPE OF EFFECT / SCOPE OF CONTROL PROBLEMS

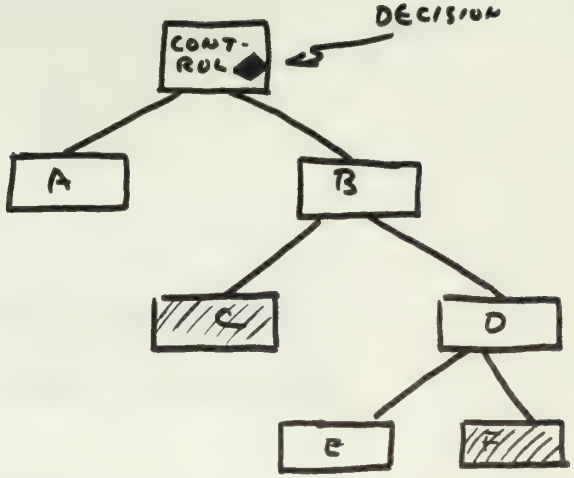
SCOPE OF CONTROL OF A MODULE
THE MODULE & ALL THOSE SUBORDINATE TO IT

SCOPE OF EFFECT OF A DECISION
ALL MODULES ULTIMATELY AFFECTED BY THE DECISION

FOR A DECISION, SCOPE OF EFFECT SHOULD BE A SUBSET OF THE SCOPE OF CONTROL OF THE MODULE IN WHICH THE DECISION IS LOCATED

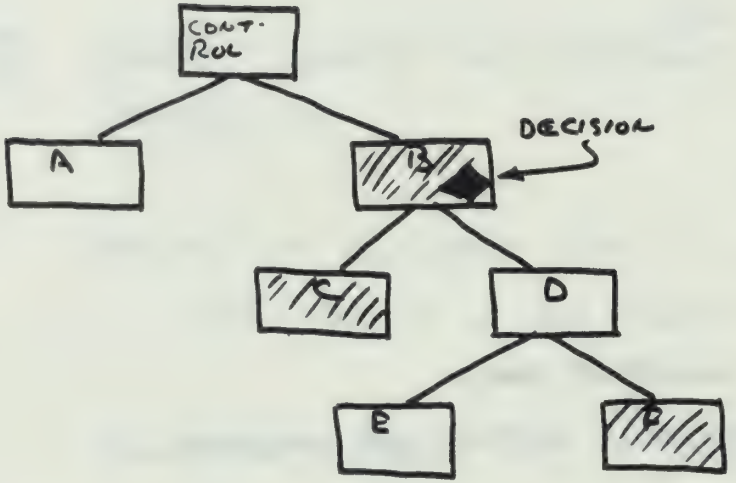


VIOLATION OF SCOPE OF EFFECT
SCOPE OF CONTROL

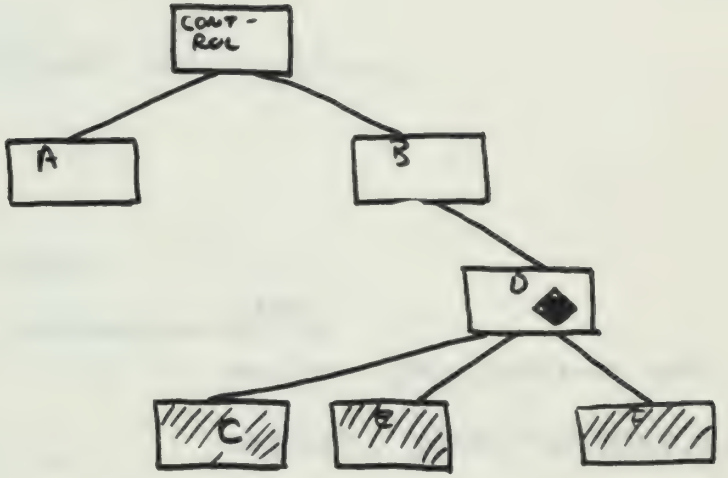


CORRECT SCOPE OF EFFECT/
CONTROL BUT DECISION TOO
HIGH

III



BETTER IMPLEMENTATION
SCOPE OF EFFECT/CONTROL



IDEAL

HEURISTIC

IV MAXIMIZE THE NUMBER OF LEVELS OF HIGHLY COHESIVE MODULES

1. BREAK MODULE INTO SMALLER & SMALLER SUB MODULES UNTIL ATOMIC LEVEL REACHED

2. BETTER TO BECOME TOO ATOMIC SINCE RECOMBINING IS ALWAYS POSSIBLE

3. BEWARE OF MODULES OF EXTREME MOLECULARITY

SYSTEM DESIGN

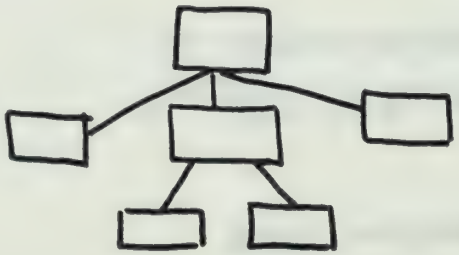
JACKSON

DEFINE DATA STRUCT.

DEVELOP PROG. STRUCT.

DEFINE TASKS

ALLOCATE TO PROG. STR.



SOME DIFFICULTY WITH COMPLEX PROBLEMS

SIMILAR

WARNIER

DEFINE DATA STRUCT.

DEVELOP PROG. STRUCT.

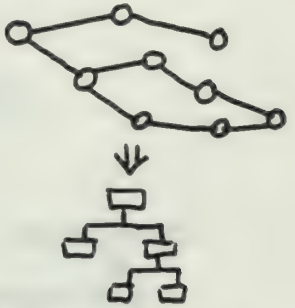
EXPAND INTO PROGRAM DETAILS

CONSTANTINE

DEFINE DATA FLOWS

DEFINE PROG. STRUC.

DEVELOP STRUCTURE CHART



ALLOWS COMPLEX DATA FLOWS

≈ SAME FOR SIMPLE DATA FLOWS

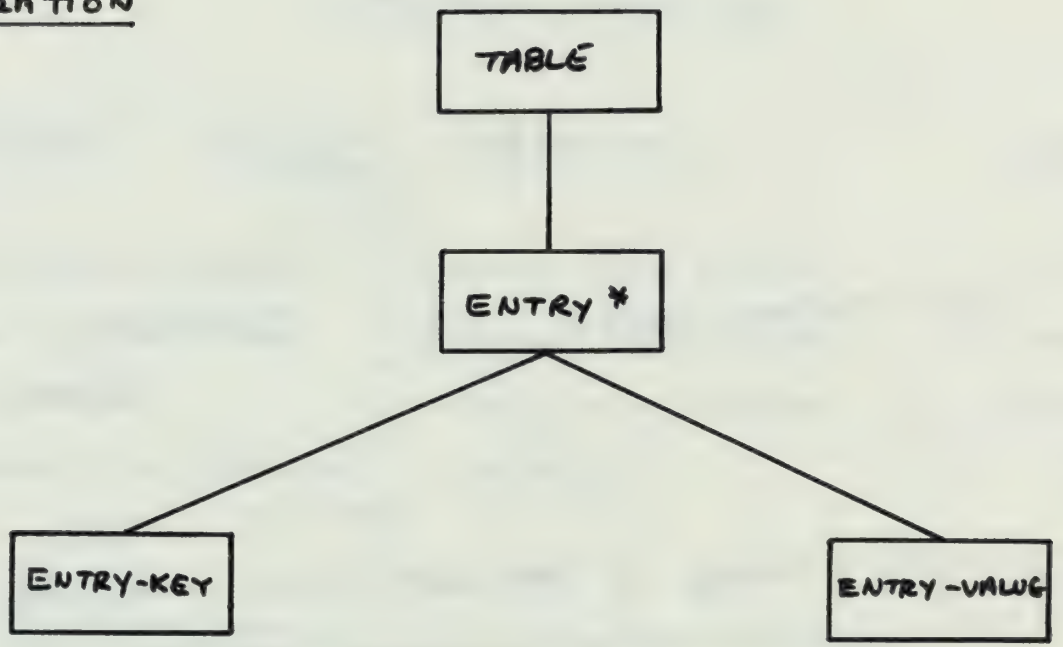
MAJOR TECHNIQUES

```
01  NUM  PIC 9(6).
01  TABLE
    02  ENTRY  OCCURS  100.
        03  ENTRY-KEY   PIC X(8).
        03  ENTRY-VALUE  PIC X(10).
```

```
PERFORM SCAN VARYING I FROM 1 BY 1
  UNTIL I > NUM.

SCAN.
  DISPLAY I , " = " , ENTRY-VALUE (I).
```

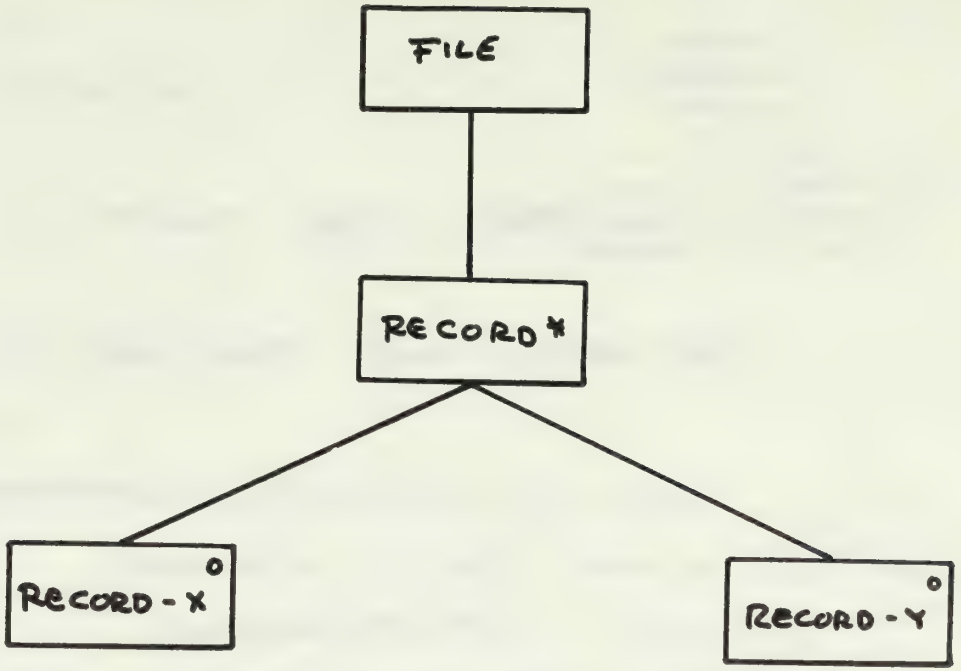
ITERATION



* INDICATES ENTRY IS MADE OF BOTH -KEY & -VALUE

SELECTION & ITERATION

JACKSON



* = ITERATION

o = EITHER - OR

PROBLEM: FIND MAX

INPUT : TABLE OF VALUES
LENGTH

OUTPUT: MAXIMUM VALUE IN THE TABLE

FUNCTION : SEARCH THE TABLE FOR THE LARGEST ENTRY VALUE

REQUIREMENT:

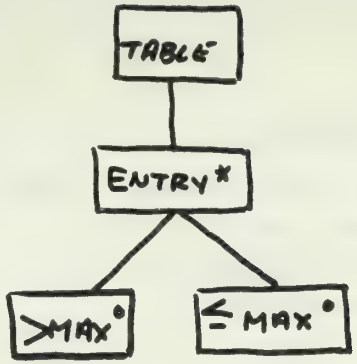
1. DRAW THE DATA STRUCTURE

2. WRITE THE CODE

FIND MAX SOLUTION

JACKSON

1. DATA STRUCTURE



2. CODE

```
MOVE TABLE(1) TO MAX.
PERFORM SEARCH VARYING I FROM 2
  BY 1 UNTIL I > LENGTH.

SEARCH.
    IF MAX < TABLE(I),
        MOVE TABLE(I) TO MAX.
        ELSE NEXT ....
```

PROBLEM "CREDIT PRINT" JACKSON

INPUT: CUSTOMER FILE

OUTPUT: CUSTOMER REPORT

FUNCTION: SELECT, PRINT CUSTOMERS HAVING "A" CREDIT RATING

1. DRAW DATA STRUCTURE

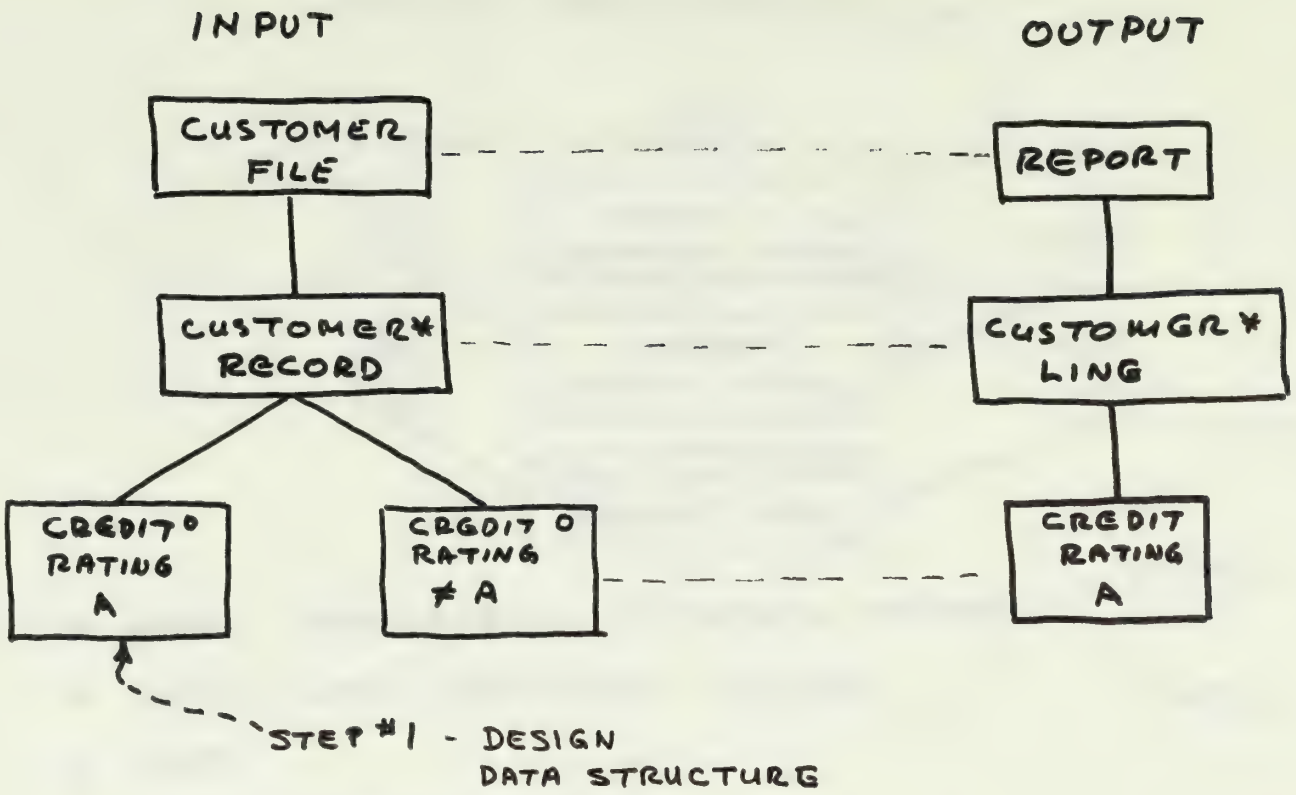
2. DESIGN PROGRAM STRUCTURE

3. LIST OPERATIONS

4. ALLOCATE OPERATIONS TO PROGRAM STRUCTURE

SOLUTION: "CREDIT PRINT"

JACKSON

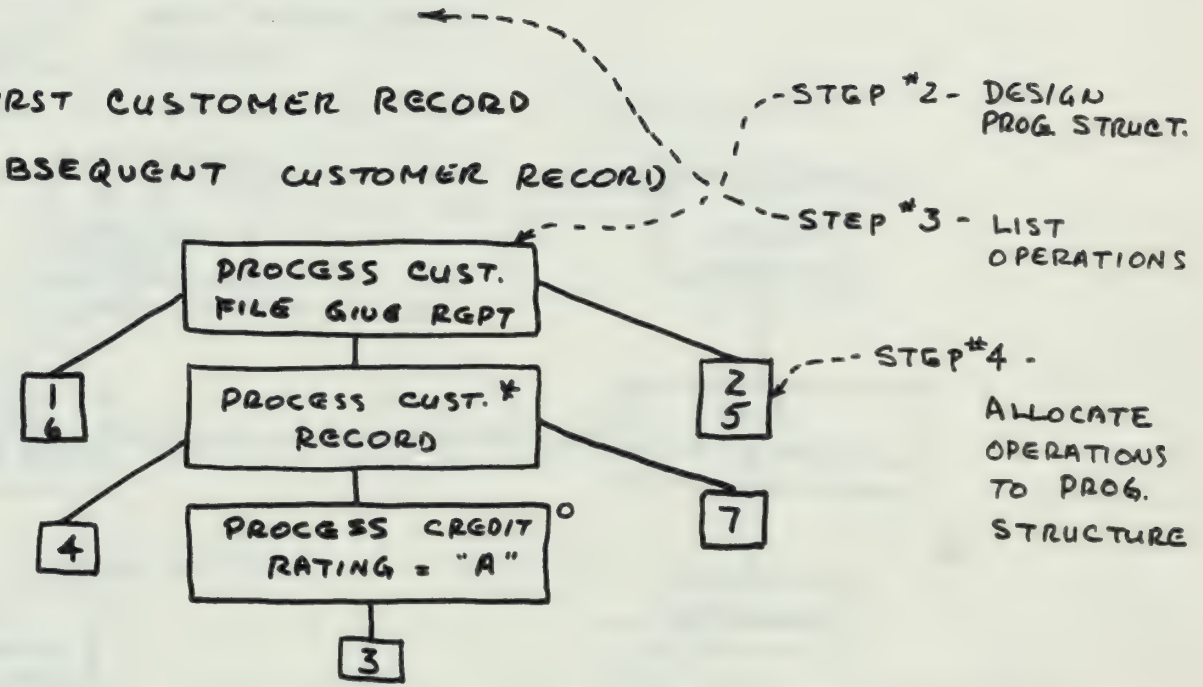


NOTE SAME LEVELS OF ABSTRACTION

5-292

LIST OF OPERATIONS "CREDIT PRINT" JACKSON

1. OPEN FILES
2. CLOSE FILES
3. DISPLAY CUSTOMER DATA
4. CHECK CREDIT RATING
5. STOP
6. READ FIRST CUSTOMER RECORD
7. READ SUBSEQUENT CUSTOMER RECORD



5-293

WARNIER'S TECHNIQUE

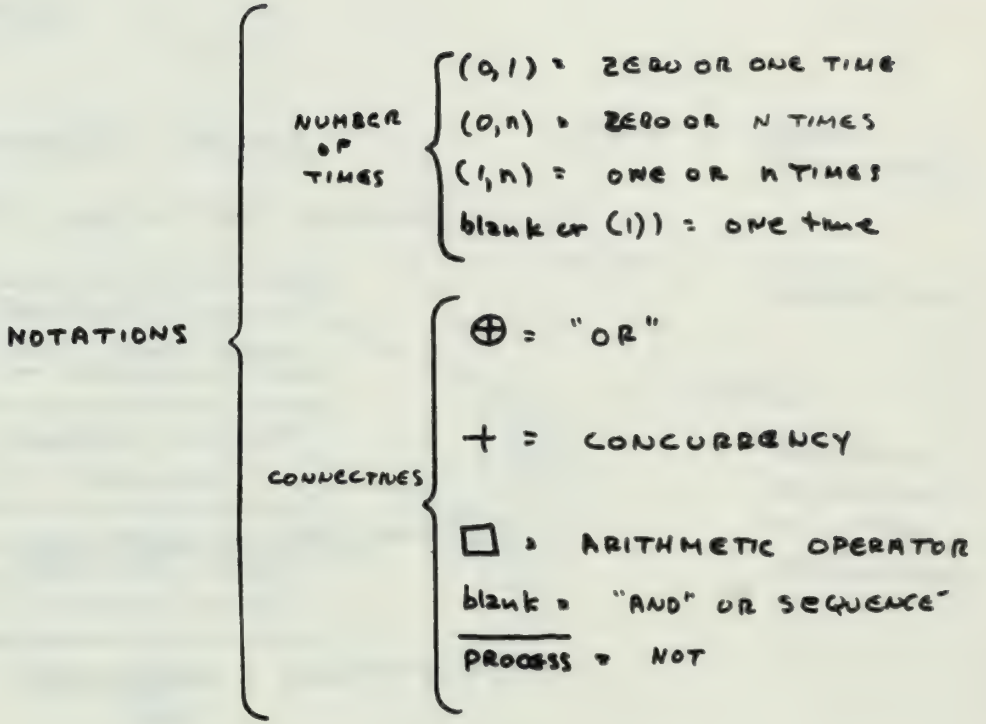
1. DEFINE DATA STRUCTURE

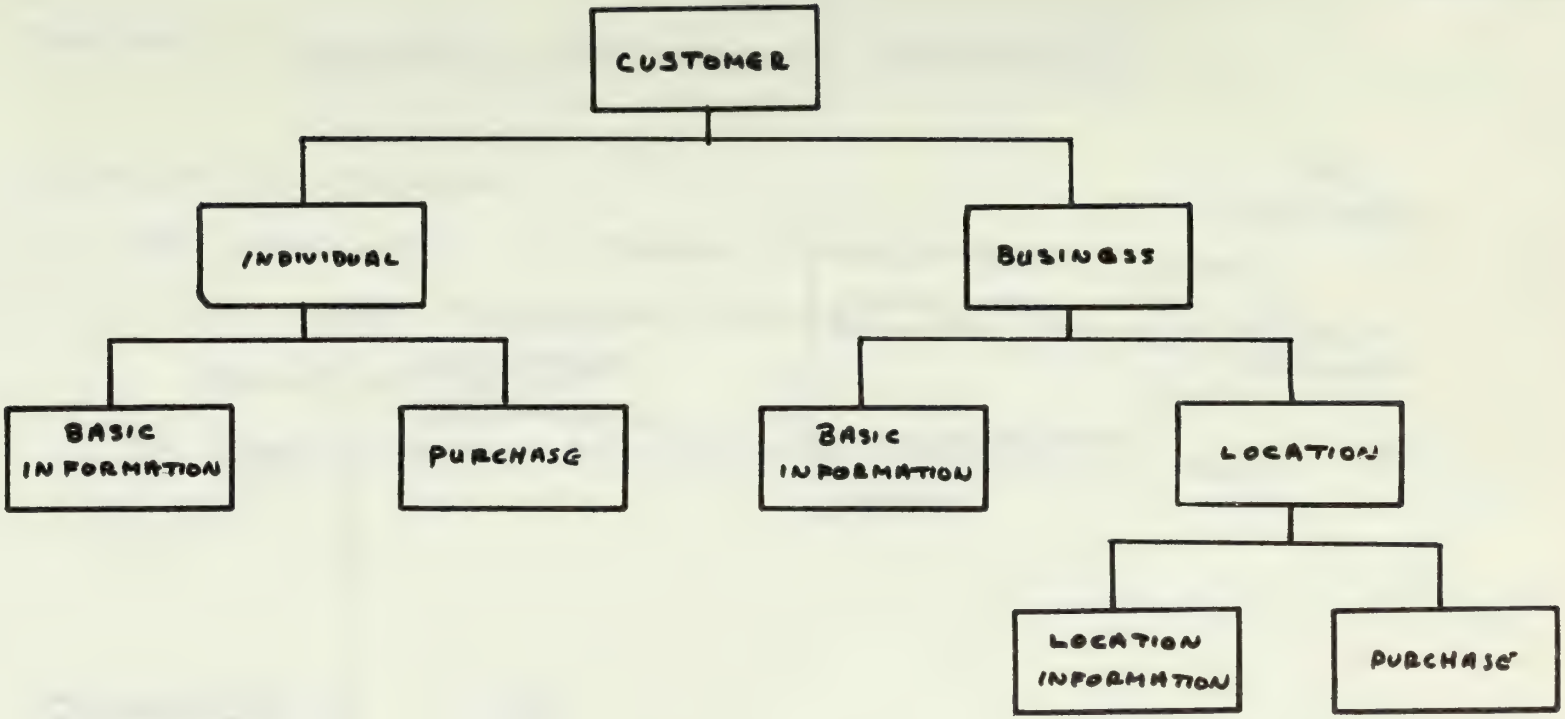
2. DEVELOP PROGRAM STRUCTURE

3. DEVELOP PROGRAM DETAILS

WARNIER DIAGRAMS

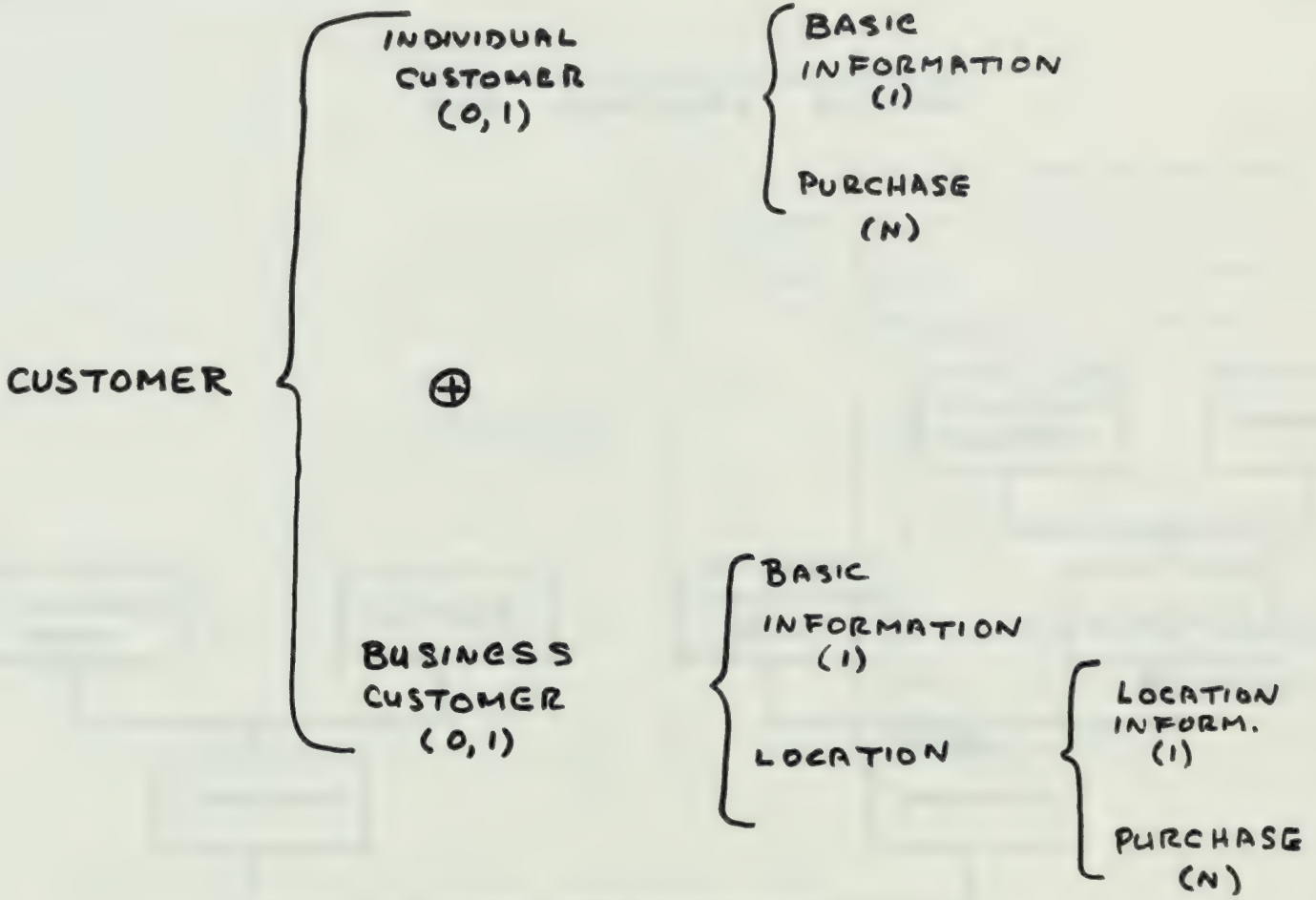
NOTATION CONVENTIONS





DATA STRUCTURE DIAGRAM

WARNIER DIAGRAM

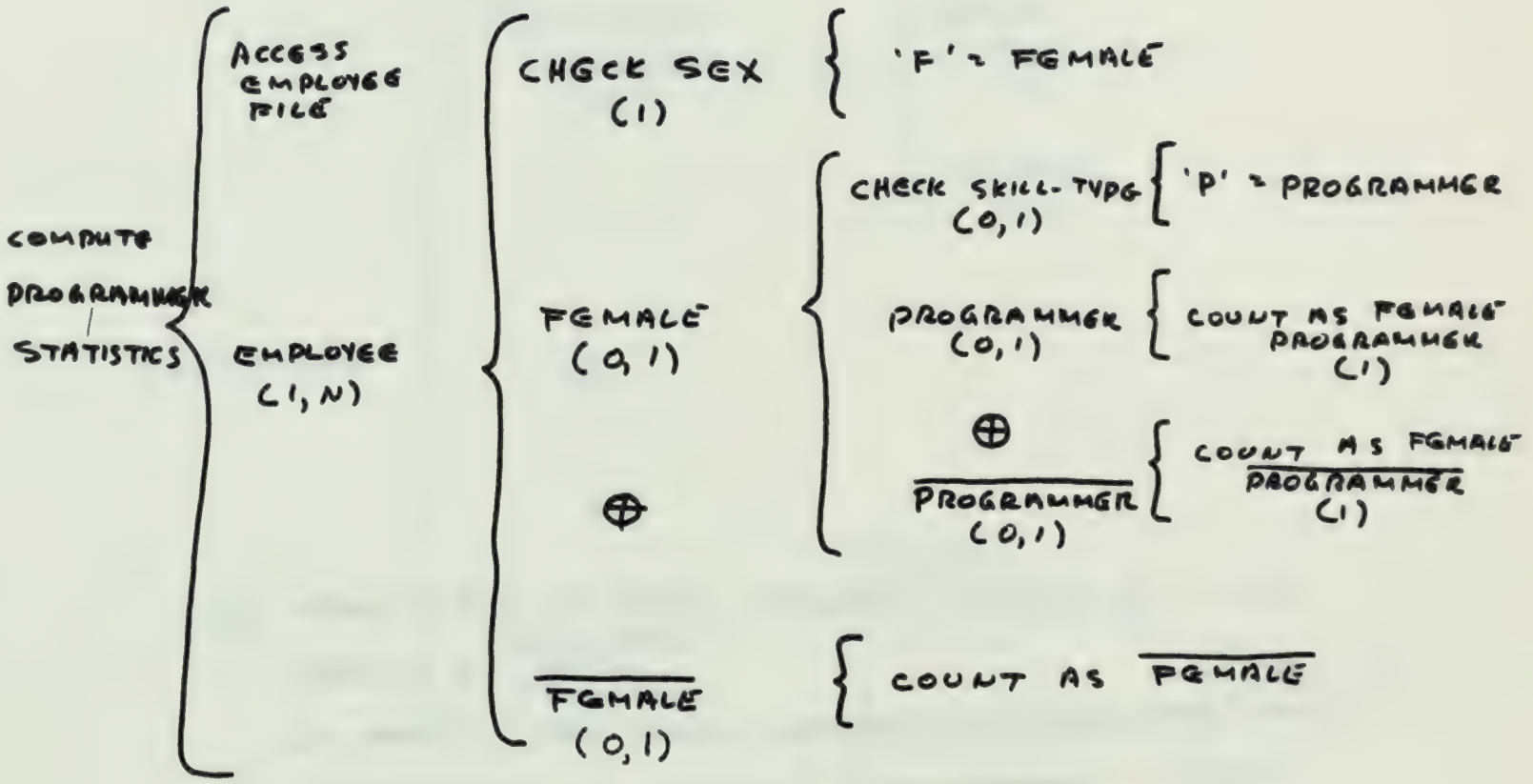


FUNCTION: COMPUTE PROGRAMMER STATISTICS

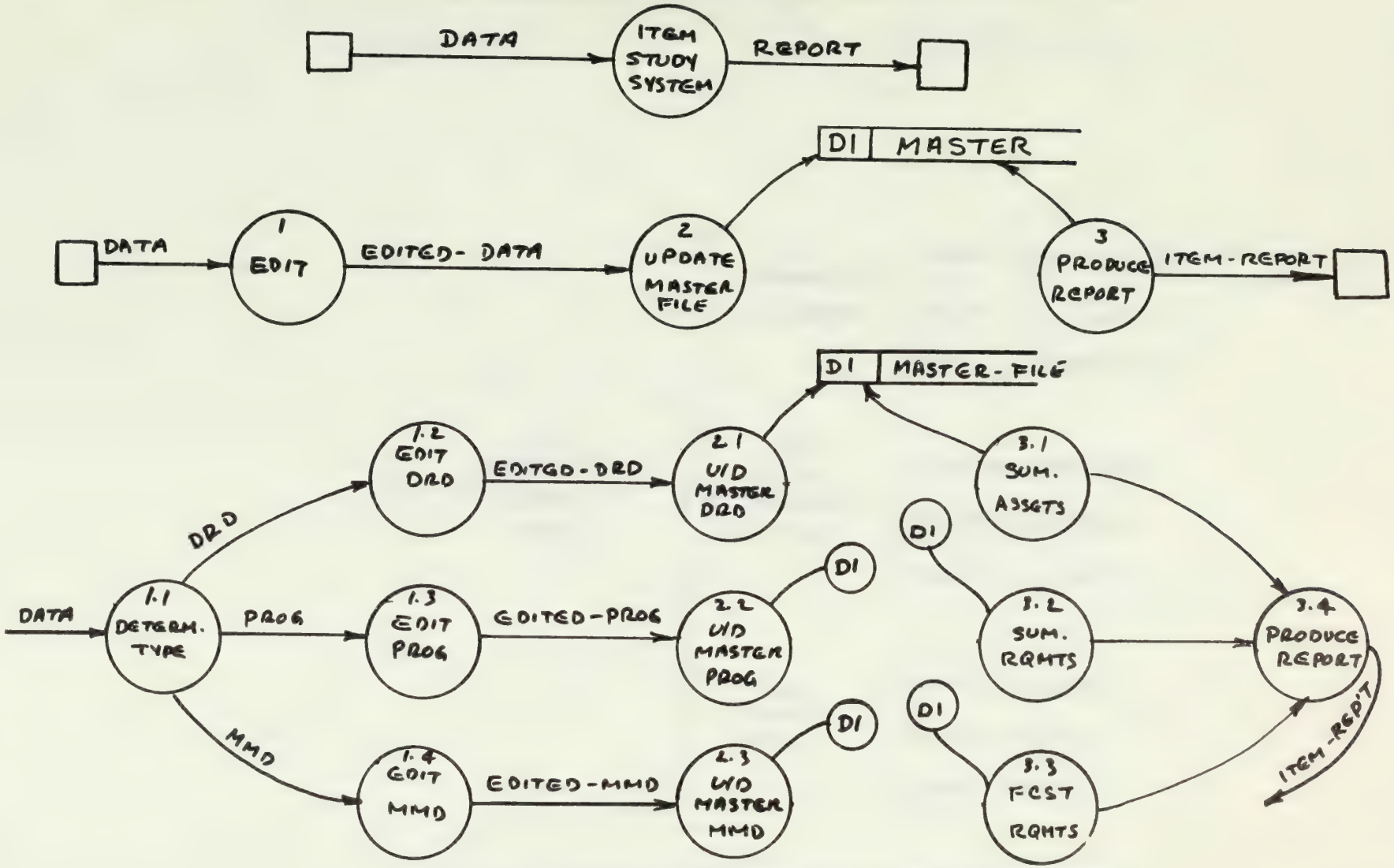
PROBLEM STATEMENT:

IF EMPLOYEE IS FEMALE, PROGRAMMER, COUNT AS SUCH. IF NON-PROGRAMMER, FEMALE, COUNT AS SUCH.

IF EMPLOYEE IS NOT FEMALE, COUNT AS SUCH.

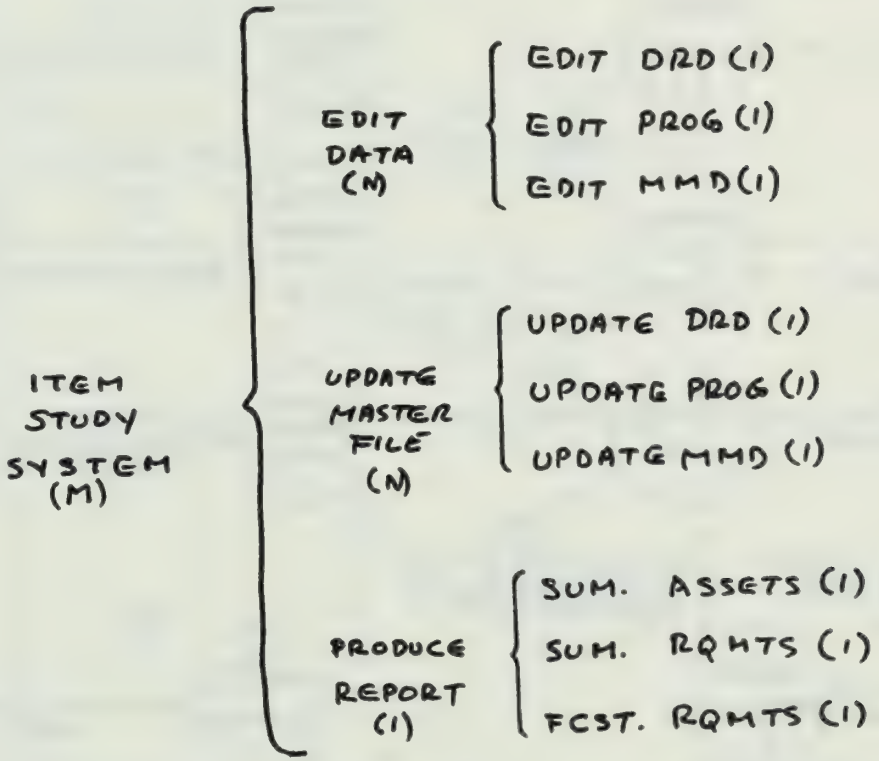

COMPUTE PROGRAMMER STATISTICS
ACCESS EMPLOYEE FILE
EMPLOYEE (1, N)
CHECK SEX (1)
'F' = FEMALE
FEMALE (0, 1)
CHECK SKILL-TYPE (0, 1)
'P' = PROGRAMMER
PROGRAMMER (0, 1)
COUNT AS FEMALE PROGRAMMER (1)
⊕
PROGRAMMER (0, 1)
COUNT AS FEMALE PROGRAMMER (1)
⊕
FEMALE (0, 1)
COUNT AS FEMALE

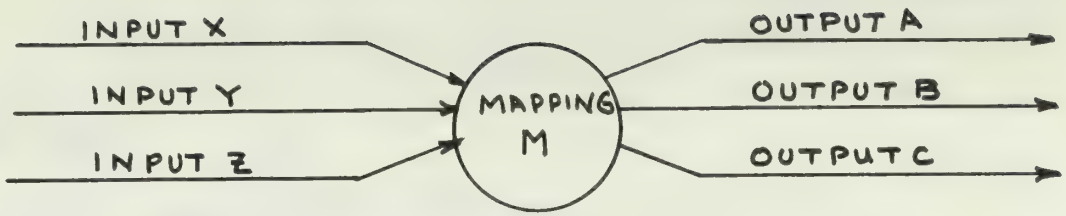
DFG — COMPARISON — WARNIER

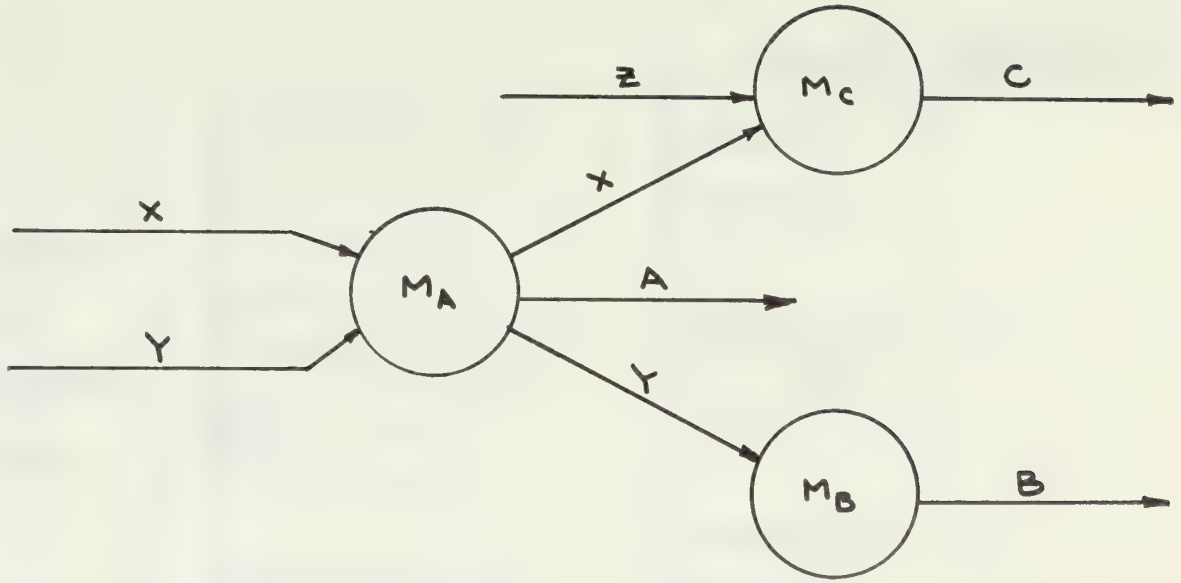


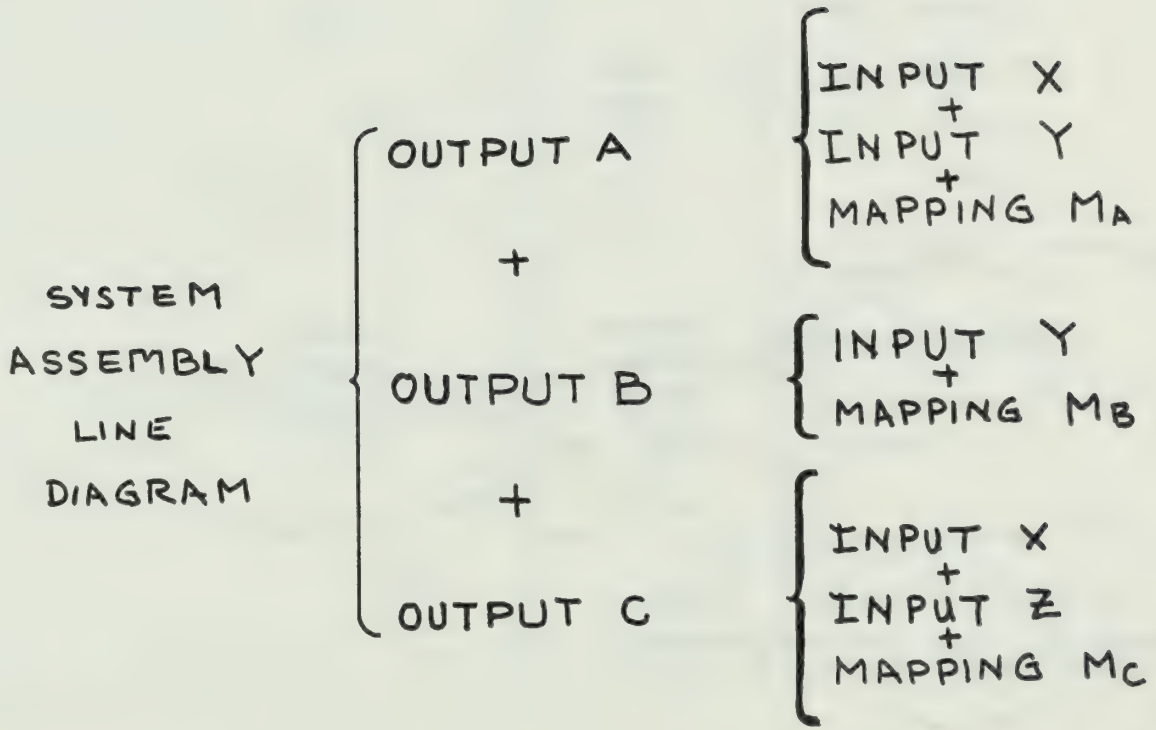
S-317A

DFG - COMPARISON - WARNIER



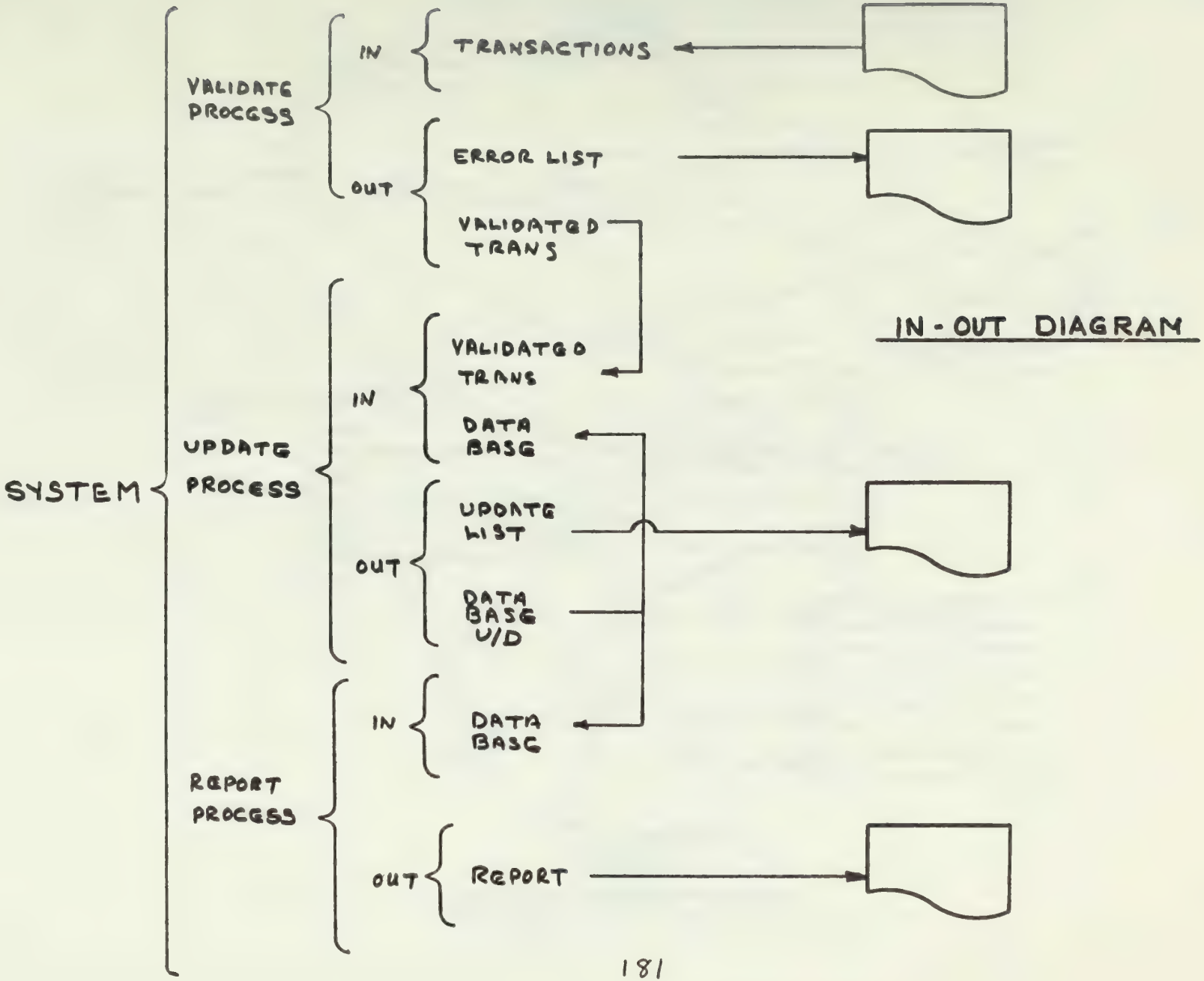

INPUT X
INPUT Y
INPUT Z
MAPPING
M
OUTPUT A
OUTPUT B
OUTPUT C


Z
M_C
C
X
X
M_A
A
Y
Y
M_B
B



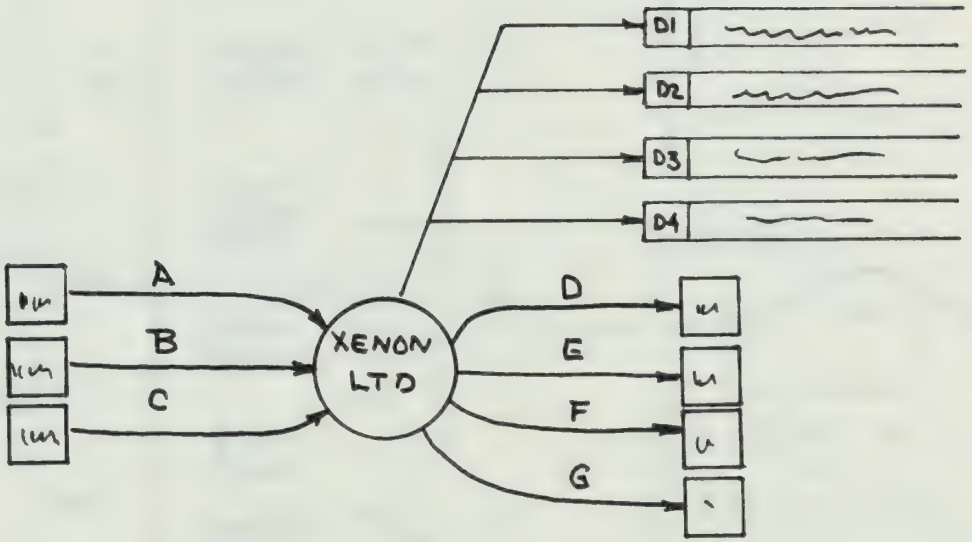
"STRUCTURED REQUIREMENTS DEFINITION"

KEN ORR 1981

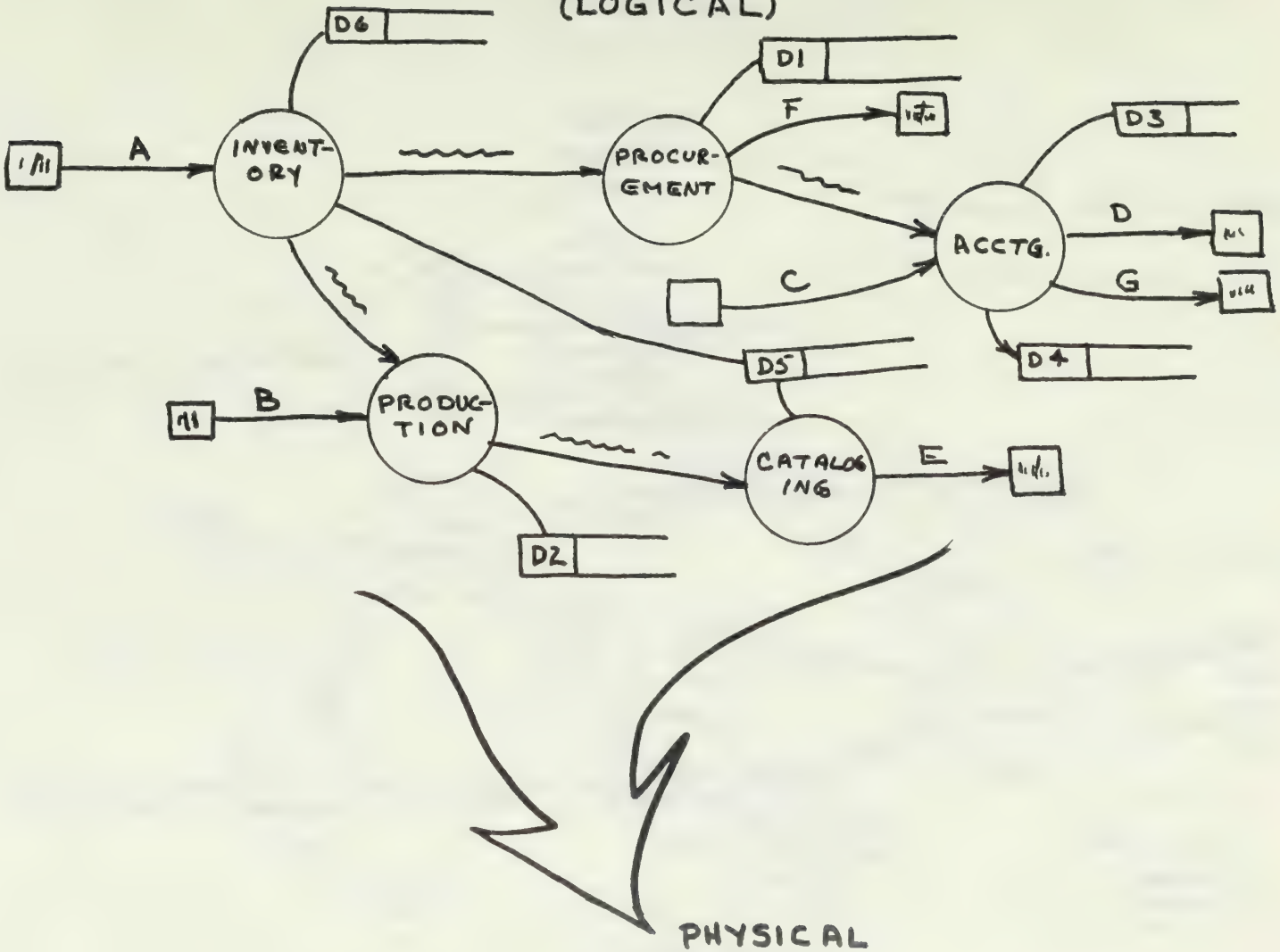


IN - OUT DIAGRAM

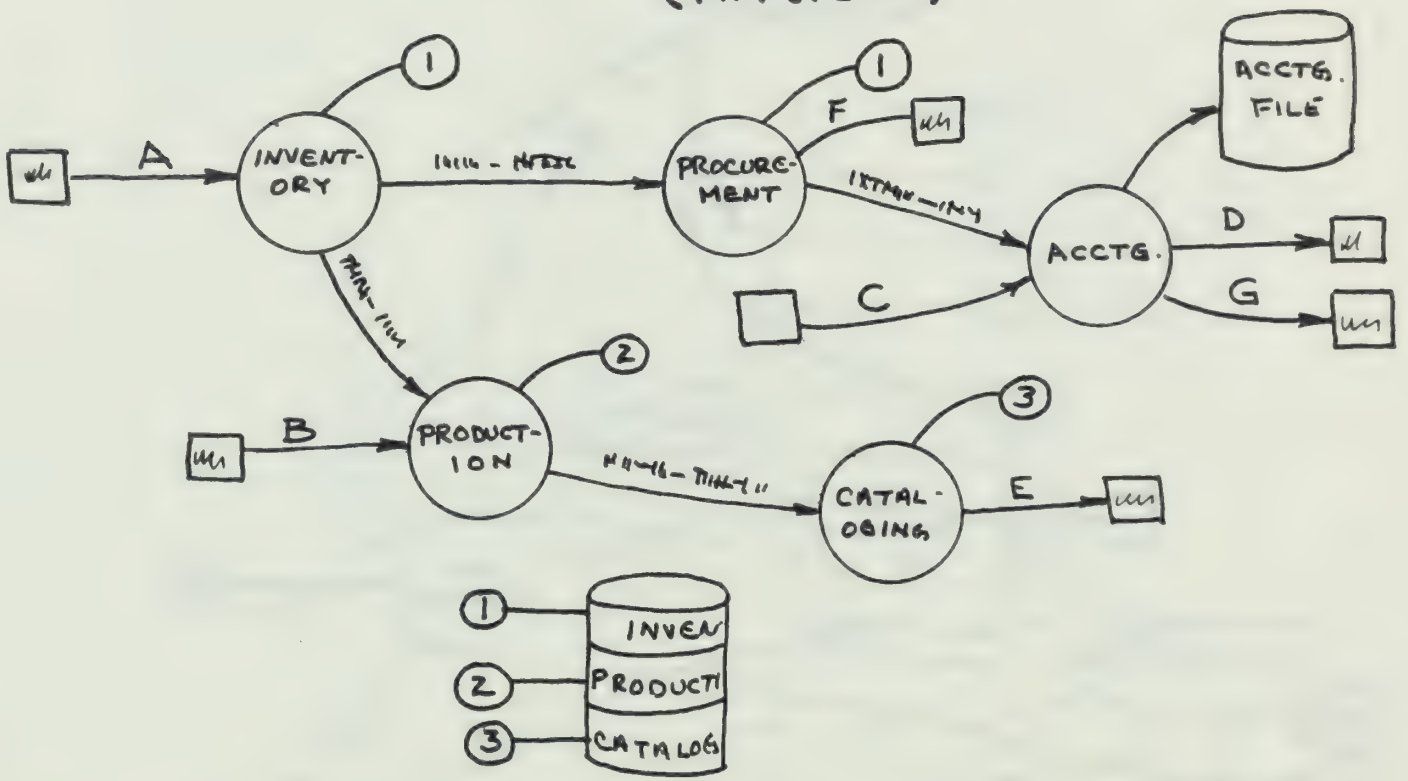
MASTER SYSTEMS CHART



GENERAL SYSTEMS CHART (LOGICAL)

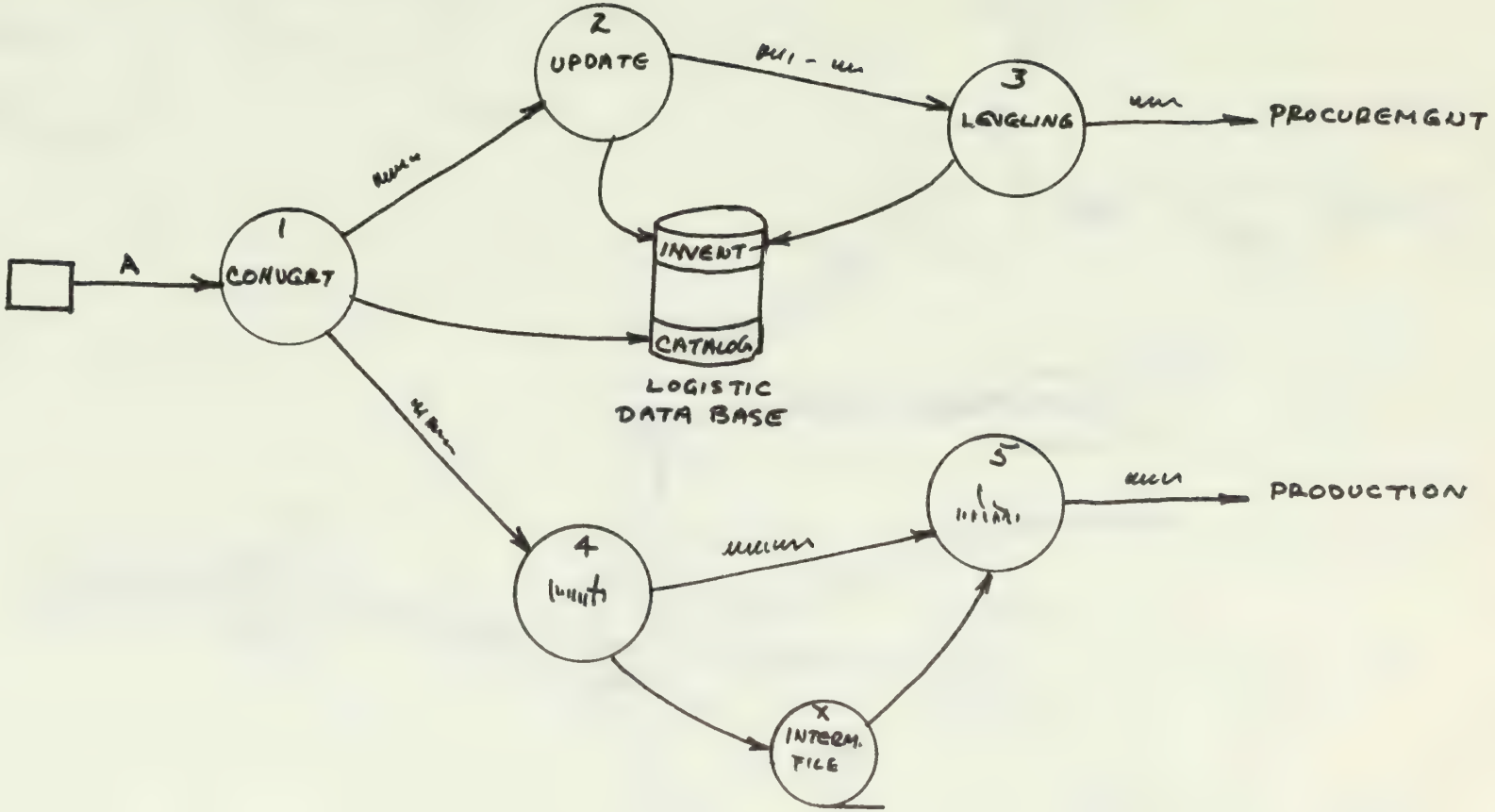


GENERAL SYSTEMS CHART (PHYSICAL)

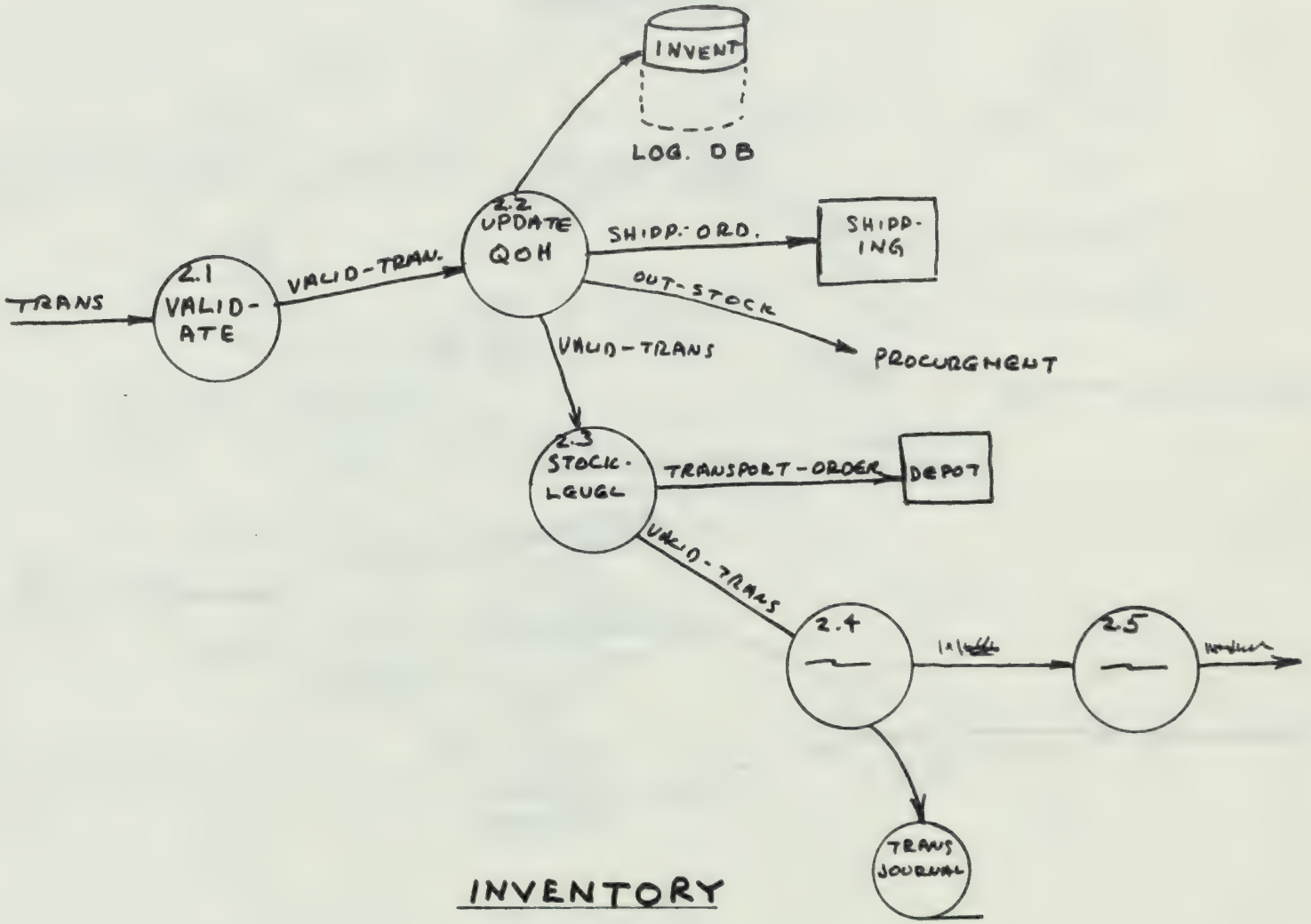


DATA BASE LOGISTIC

APPLICATION CHART



INVENTORY

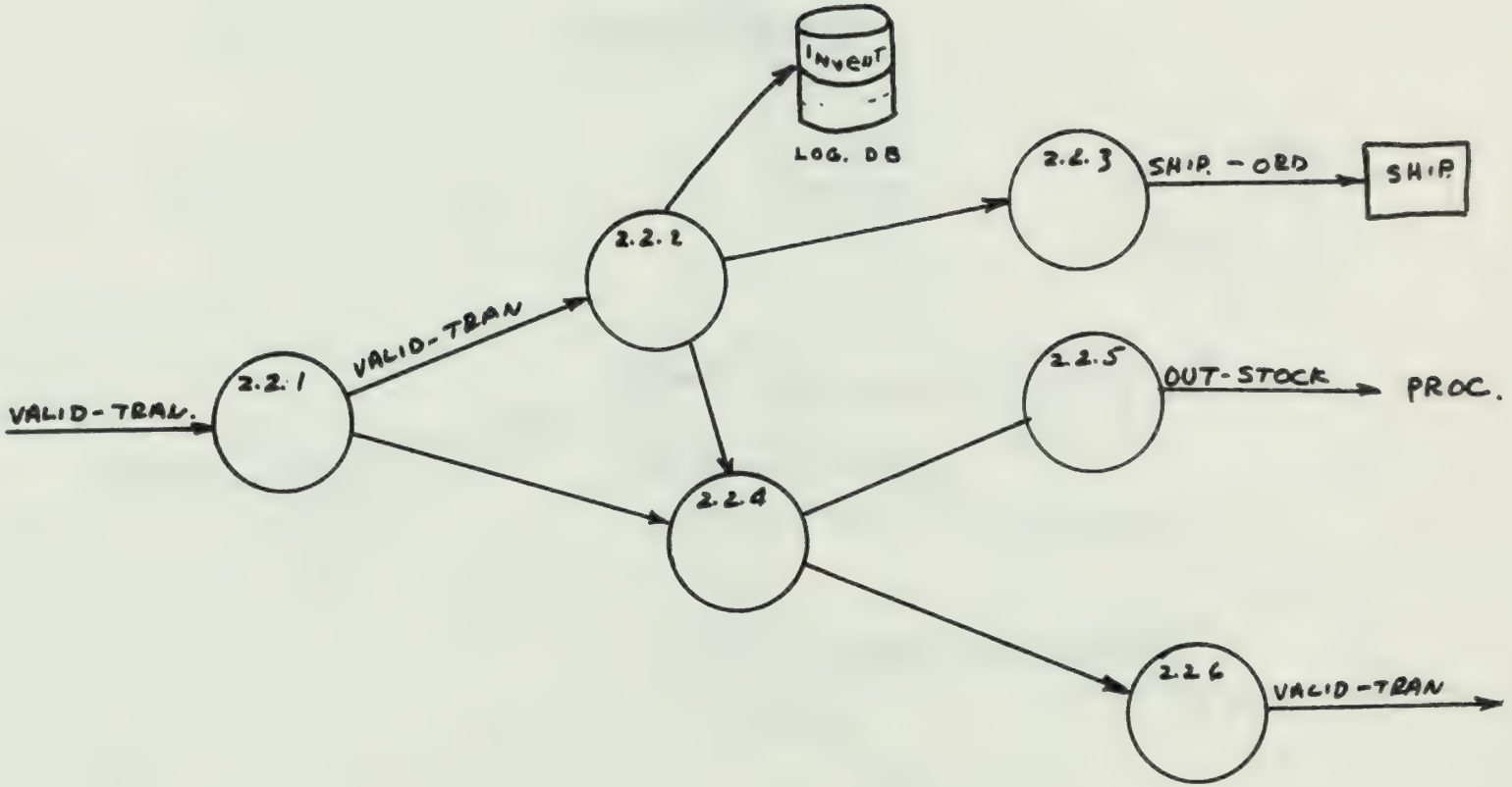


INVENTORY

(UPDATE)

MINI-SPECS.

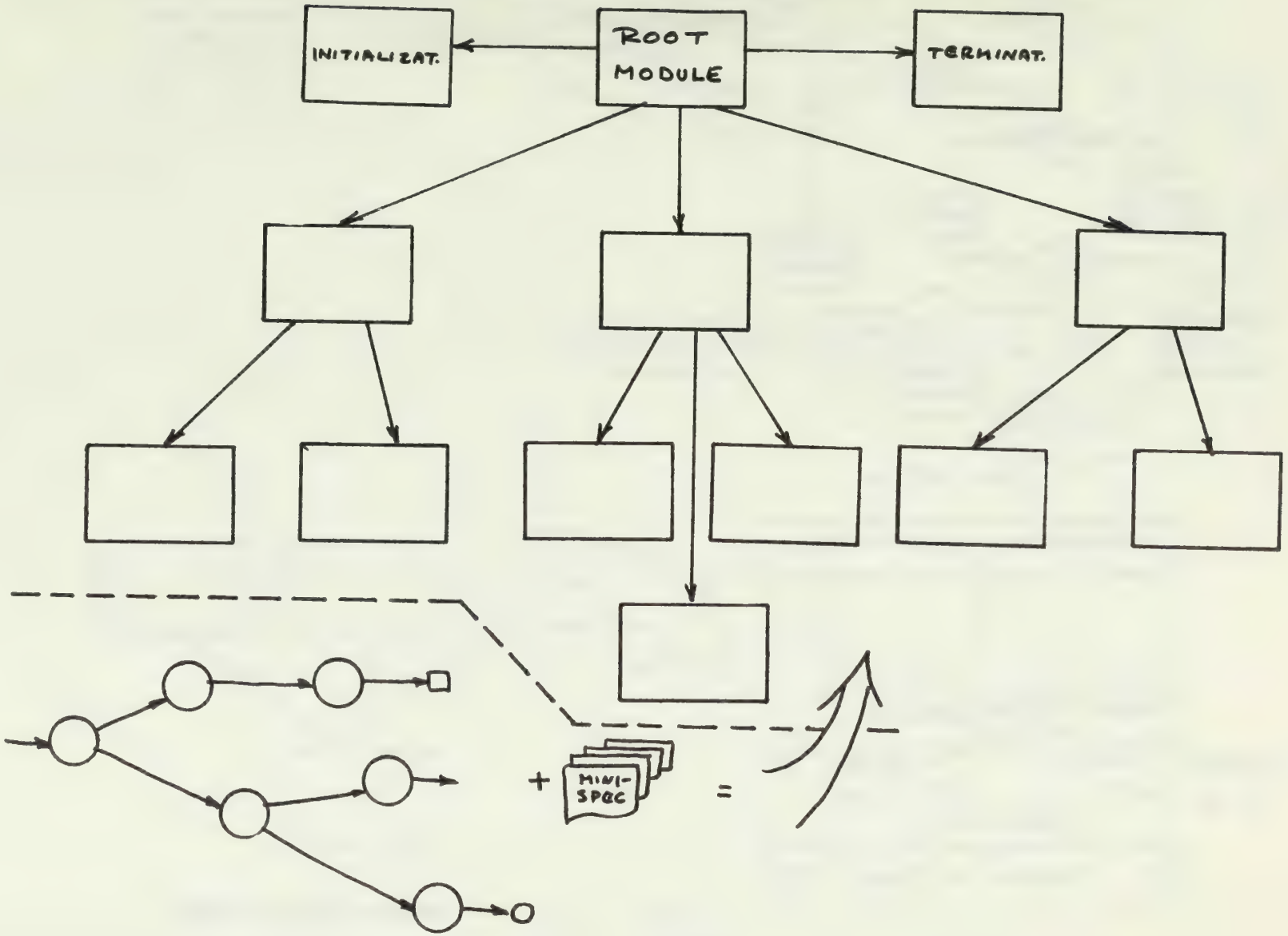
5-313A



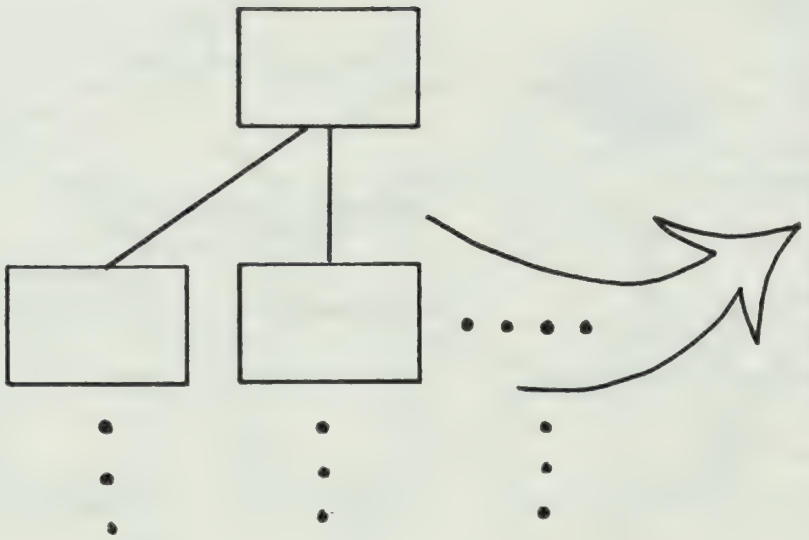
INVENTORY

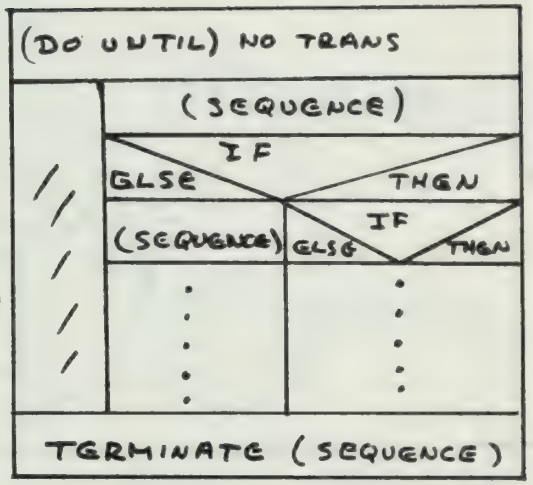
5-3188

STRUCTURE CHART



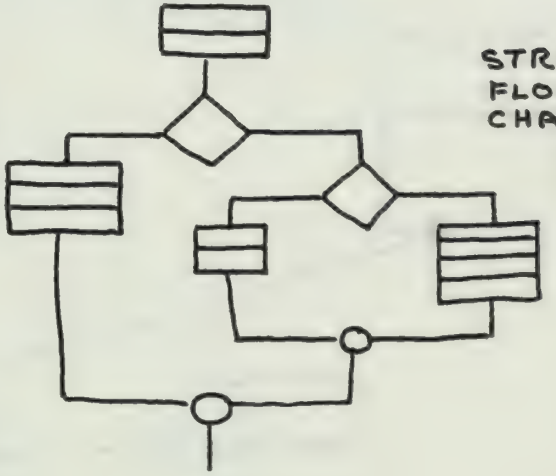
STRUCTURE CHART





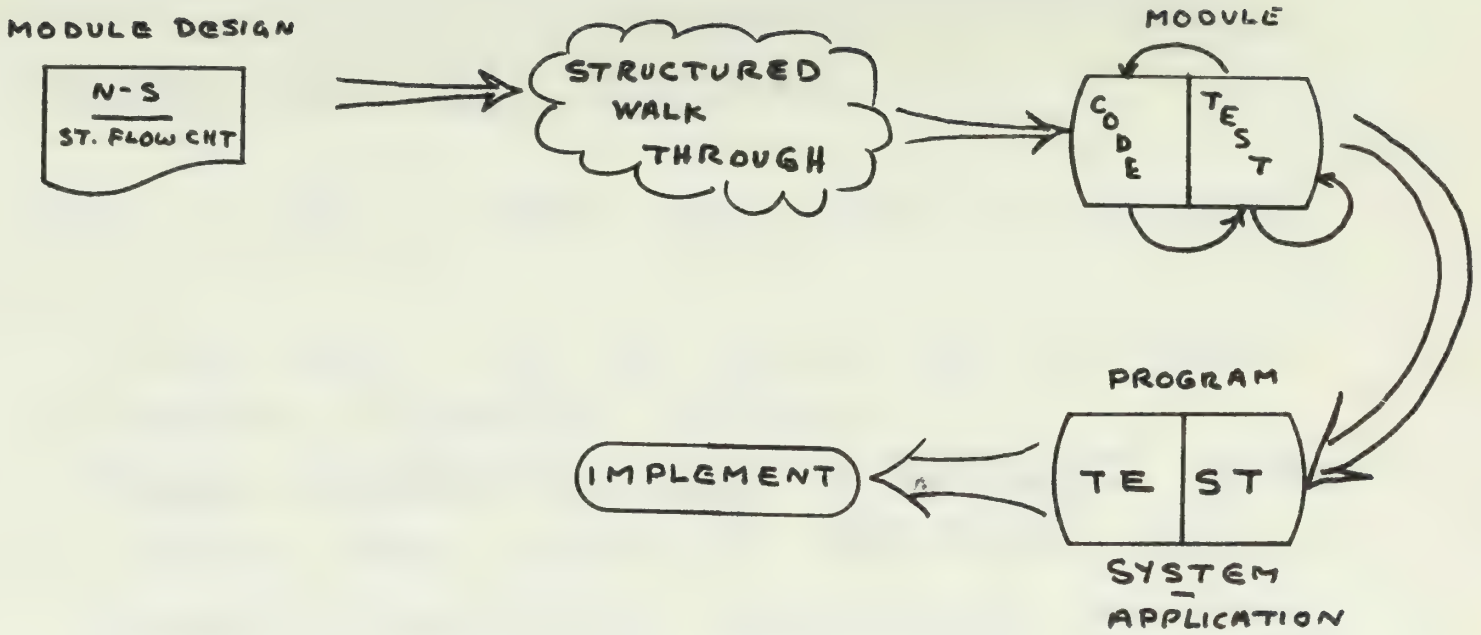
N-S CHART

— OR —



STRUCT. FLOW CHART

5-315


MODULE DESIGN
N-S
ST. FLOW CHT
STRUCTURED
WALK
THROUGH
MODULE
CODE
TEST
PROGRAM
TE
ST
IMPLEMENT
SYSTEM
APPLICATION

TRANSFORM ANALYSIS

A STRATEGY FOR DESIGNING HIGHLY MODULAR PROGRAMS & SYSTEMS

BASIC IDEA IS TO DESIGN A PROGRAM SO THAT THE MAIN PROGRAM CALLS FOR HIGHLY PROCESSED LOGICAL DATA. EVENTUALLY GENERATES LOGICAL OUTPUT DATA.

LOWER LEVEL MODULES TRANSFORM PHYSICAL INPUT TO LOGICAL INPUT & TRANSFORM LOGICAL OUTPUT TO PHYSICAL OUTPUT.

TRANSFORM ANALYSIS STRATEGY

STEP 1. DRAW DATA FLOW GRAPH OF PROBLEM

WORK FROM INPUT TO OUTPUT OR VICE VERSA

DO NOT SHOW CONTROLS

IGNORE INITIALIZATION, TERMINATION (PERPETUAL MOTION)

OMIT SIMPLE ERROR PATHS

LABEL DATA PASSING & FUNCTIONS CAREFULLY

STEP 2. IDENTIFY EXTERNAL "CONCEPTUAL" STREAMS OF DATA — DATA INDEPENDENT OF PHYSICAL DEVICES

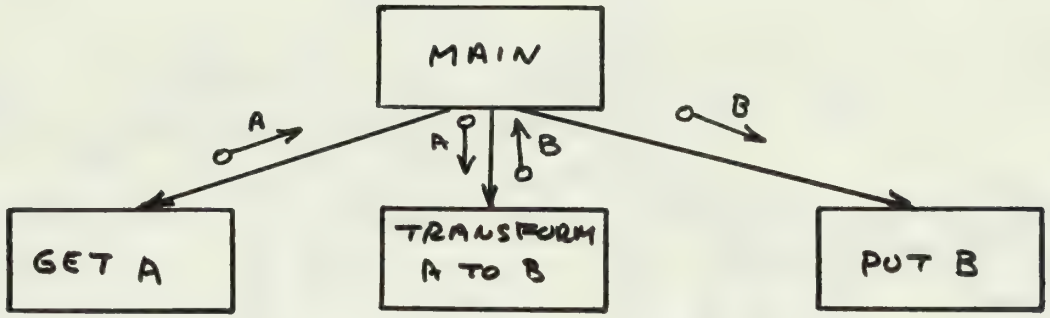
STEP 3. IDENTIFY MAJOR AFFERENT, EFFERENT STREAMS

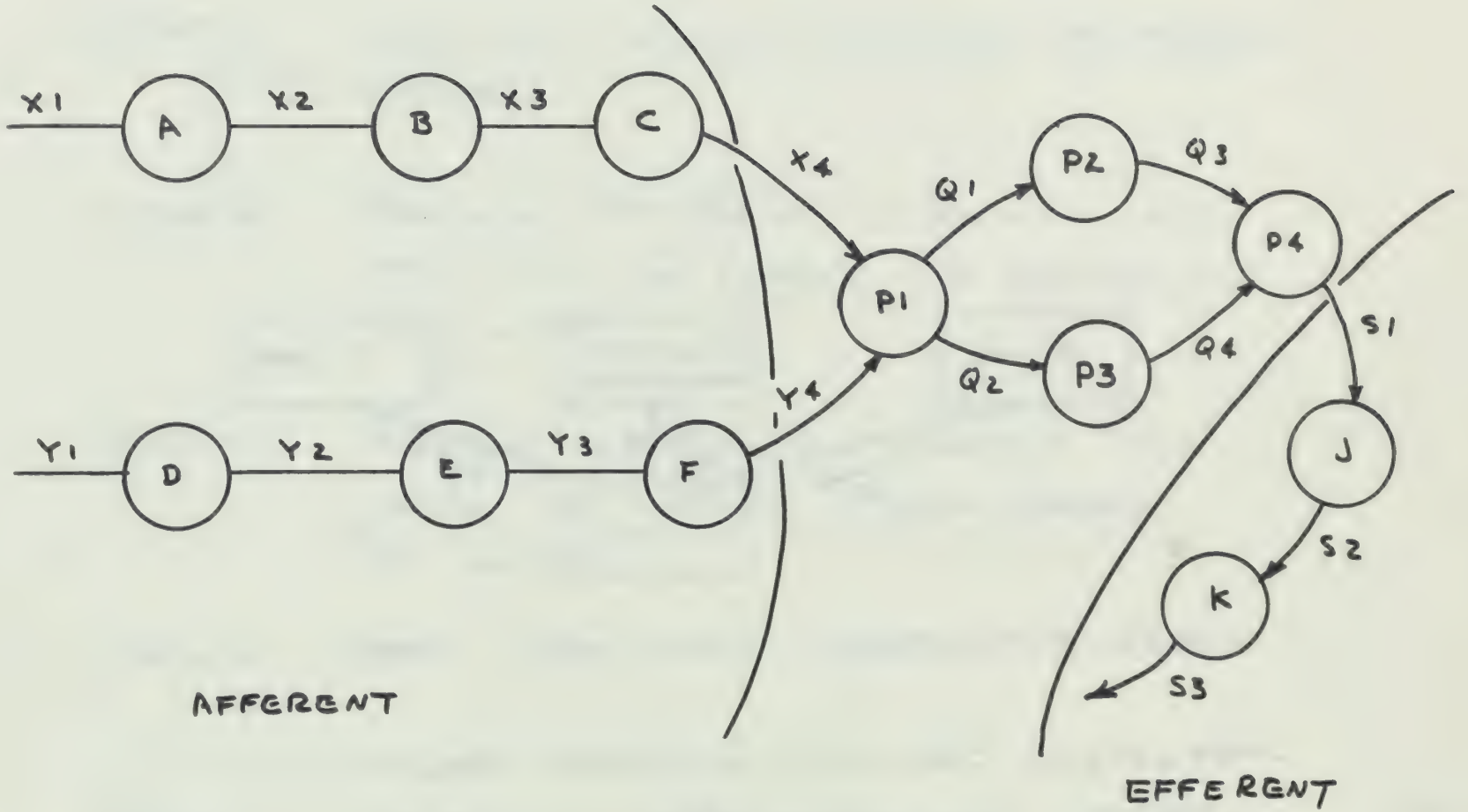
STEP 4. FOLLOW AFFERENT STREAMS UNTIL THEY CAN NO LONGER BE IDENTIFIED AS INPUT

STEP 5. FOLLOW EFFERENT STREAMS BACK UNTIL NO LONGER RECOGNIZABLE AS OUTPUT

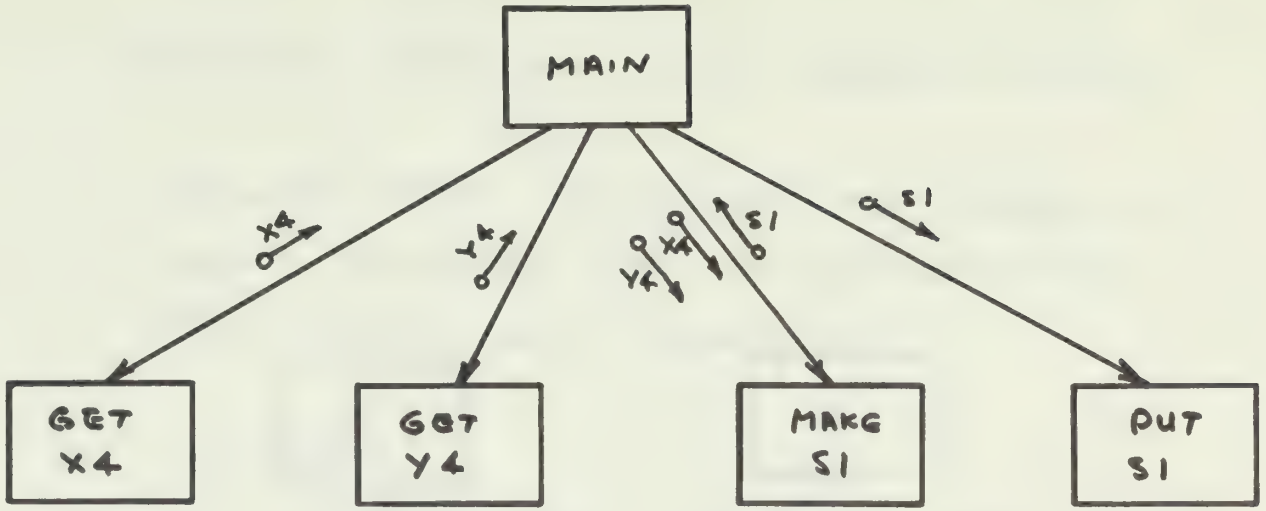
STEP 6. IDENTIFY TRANSFORMATION "BUBBLES" IN THE MIDDLE - CENTRAL TRANSFORMS

STEP 7. DRAW TOP LEVEL STRUCTURE CHART




X1
A
X2
B
X3
C
X4
Y1
D
Y2
E
Y3
F
Y4
P1
Q1
P2
Q3
P4
Q2
P3
Q4
S1
J
S2
K
S3
AFFERENT
EFFERENT

XENON CORP.



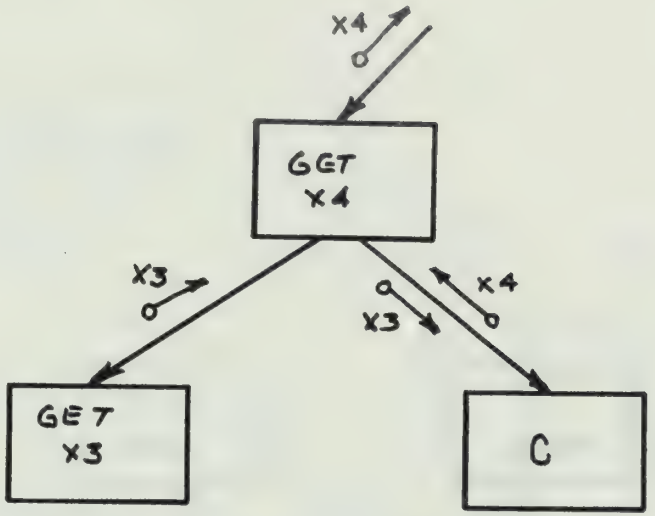
TOP LEVEL

STRUCTURE CHART

197

5-30F

FACTORING



AT EACH STEP IN THE DESIGN PROCESS

1. FACTOR MODULE INTO SUBORDINATES

2. ASSURE THAT SUBORDINATE MODULES ARE CAPABLE OF IMPLEMENTING THE REQUIRED FUNCTION

3. CHECK :

   SUBORDINATE MODULE COHESIVENESS
   SCOPE OF EFFECT/CONTROL CONFLICTS
   COUPLING
   MODULE SIZE
   EXCESSIVE FAN-OUT

5-307

☆ U.S. GOVERNMENT PRINTING OFFICE: 1982—556-176